

UNIVERSIDADE DO EXTREMO SUL CATARINENSE - UNESC
CURSO DE CIÊNCIA DA COMPUTAÇÃO

JOÃO PAULO NIEHUES WESSLER

**AUTOMATIZAÇÃO E MONITORAMENTO DE PROCESSOS RELACIONADOS AO
BEM-ESTAR ANIMAL DENTRO DA CADEIA PRODUTIVA DA SUINOCULTURA
POR MEIO DE MICROCONTROLADORES INTEGRADOS A UMA APLICAÇÃO
MOBILE**

CRICIÚMA
2019

JOÃO PAULO NIEHUES WESSLER

**AUTOMATIZAÇÃO E MONITORAMENTO DE PROCESSOS RELACIONADOS AO
BEM-ESTAR ANIMAL DENTRO DA CADEIA PRODUTIVA DA SUINOCULTURA
POR MEIO DE MICROCONTROLADORES INTEGRADOS A UMA APLICAÇÃO
MOBILE**

Trabalho de Conclusão de Curso, apresentado
para obtenção do grau de Bacharel no curso de
Ciência da Computação da Universidade do
Extremo Sul Catarinense, UNESC.

Orientador: Prof. Me. Paulo João Martins

CRICIÚMA

2019

JOÃO PAULO NIEHUES WESSLER

**AUTOMATIZAÇÃO E MONITORAMENTO DE PROCESSOS RELACIONADOS AO
BEM-ESTAR ANIMAL DENTRO DA CADEIA PRODUTIVA DA SUINOCULTURA
POR MEIO DE MICROCONTROLADORES INTEGRADO A UMA APLICAÇÃO
MOBILE**

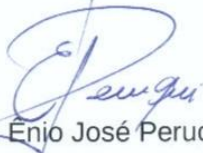
Trabalho de Conclusão de Curso aprovado pela Banca Examinadora para obtenção do Grau de Bacharel, no Curso de Ciência da Computação da Universidade do Extremo Sul Catarinense, UNESC, com Linha de Pesquisa em Comunicação de Dados.

Criciúma, 24 de Junho de 2019.

BANCA EXAMINADORA



Prof. Me. Paulo João Martins - UNESC - Orientador



Prof. Esp. Enjo José Peruchi - UNESC



Prof. Me. Ramon Venson - CEDUP

**Dedico este trabalho aos meus pais, que são
minha fonte de inspiração e sempre me
forneceram todo tipo de apoio para que meus
objetivos fossem alcançados**

AGRADECIMENTOS

Primeiramente agradeço a Deus, por proporcionar inúmeras bençãos e fornecer a sabedoria necessária para percorrer os caminhos corretos, afim de alcançar meus objetivos e concluir mais esta etapa da minha vida.

Aos meus pais Jairton Wessler e Isabel Niehues Wessler, que sempre prestaram um apoio incondicional em todos os momentos, ao qual mostram-me o quanto é importante batalhar por algo que se deseja. A eles todo o meu amor e agradecimento.

A minha namorada Vitória Speck Dutra, que foi uma das principais pessoas a acreditar no meu potencial durante este período e que sempre prestou todo tipo de apoio em tudo o que fosse necessário, principalmente me dando forças e fazendo com que eu persistisse na luta pelos meus objetivos. A ela todo o meu amor e carinho.

Um agradecimento especial para o meu amigo Tiago Behenck dos Santos, este irmão de outra mãe, que esteve junto a mim desde o início da faculdade, onde juntos conseguimos enfrentar todas as dificuldades através de uma parceria forte. A ele minha eterna gratidão e que essa amizade permaneça para toda a vida.

A todos os meus amigos em geral, que me deram forças e apoio para a conclusão desta etapa, sou muito grato a todos vocês.

Um agradecimento especial ao meu orientador Paulo João Martins, que não mediu esforços para me auxiliar em tudo o que fosse necessário para que eu conseguisse obter sucesso nesta etapa. A ele meu eterno agradecimento.

A UNESC e a todos os professores do curso de Ciência da Computação, que proporcionaram um ótimo ambiente para o aprendizado, bem como profissionais qualificados e dedicados na missão de repassar os seus conhecimentos a todos os alunos. A todos vocês meu muito obrigado.

**“Não há nada que dominemos inteiramente a
não ser os nossos pensamentos.”**

René Descartes

RESUMO

É notório que o uso da tecnologia está cada vez mais presente no cotidiano das pessoas e com um nível de participação sendo exponencialmente utilizado nos ambientes de produção, ao qual progressivamente proporciona melhorias e otimiza todo tipo de atividade. No cenário de produção de suínos não é diferente, onde a contínua implementação tecnológica, proporciona cada vez mais o aumento da produtividade e o auxílio no manejo dos animais. Porém, infelizmente esta realidade não está presente em todos os locais de produção no território brasileiro, com isso, ainda tendo muito a melhorar principalmente em pequenas e médias propriedades produtoras. Diante deste contexto, o presente trabalho visa a pesquisa e o desenvolvimento de um protótipo para o monitoramento das variáveis de temperatura, umidade, frequência sonora e luminosidade, relacionados aos níveis de bem-estar animal na suinocultura. Para tal, foi realizado a montagem de uma simulação em forma de maquete, utilizando-se do microcontrolador Arduino integrado a sensores específicos para cada variável monitorada, ao qual por intermédio deste, os dados são coletados e enviados para um servidor na nuvem. Posteriormente para o gerenciamento dessas informações, foi realizado o desenvolvimento de uma aplicação mobile, utilizando-se do *framework* Ionic, ao qual através desta, o usuário pode gerenciar todo o fluxo de dados obtidos, com isso, podendo também realizar o controle do clima no ambiente, por meio do acionamento de equipamentos acoplados a maquete, simulando assim o resfriamento e o aquecimento da mesma. Por fim, obteve-se um protótipo de sistema de monitoramento, que pode ser aplicado a um ambiente de produção de suínos, de modo a otimizar o manejo e gerar melhores índices de produtividade.

Palavras-chave: Suinocultura. Bem-estar animal. Microcontroladores. Monitoramento. *Framework* Ionic.

ABSTRACT

It is notorious that the use of technology is being more present in people's daily lives and in a level of participation being exponentially used in production environments, which progressively provides improvements and optimizes all types of activity. In the scenario of pig production is no different, where the continuous technological implementation, provides increased productivity and assistance in the management of animals. However, unfortunately this reality is not present in all the locals productions in the Brazilian territory, with this, still having much to improve mainly in small and medium producing properties. In this context, the present work aims at the research and development of a prototype for the monitoring of variables of temperature, humidity, sound frequency and luminosity related to animal welfare levels in swine. For that, a simulation was build in a model, using the Arduino microcontroller integrated with sensors specific to each monitored variable, through which the data is collected and sent to a server in the cloud. For the management of the informations, was development a mobile application using the framework Ionic, which the user can manage all the data flow obtained and control the climate in the environment by the activation of equipment coupled to the model, simulating the cooling and heating. Finally, a prototype of a monitoring system was obtained, which can be applied to a pig production environment, in order to optimize management and generate better productivity indexes.

Keywords: Swine. Animal welfare. Microcontrollers. Monitoring. Framework Ionic.

LISTA DE ILUSTRAÇÕES

Figura 1 - Ranking de produção mundial de suínos em 2018	21
Figura 2 - Ranking de produção de suínos no território brasileiro em 2018	21
Figura 3 - Arquitetura do sistema operacional Android	30
Figura 4 - Arquitetura do sistema operacional IOS.....	32
Figura 5 - Resumo de lançamentos de aplicativos por plataforma.....	33
Figura 6 - Comparativo entre <i>native apps</i> , <i>webapps</i> e aplicações híbridas	34
Figura 7 - Resumo de ferramentas necessários para a execução do Ionic.....	37
Figura 8 - Formas de energia em um sensor.	40
Figura 9 - Arduino Mega2560.....	43
Figura 10 - Sensor LM35.....	43
Figura 11 - Sensor HR202L.....	44
Figura 12 - Sensor AM2302	44
Figura 13 - Sensor DHT11	45
Figura 14 - Sensor KY-038.....	46
Figura 15 - Sensor BH1750FVI	46
Figura 16 - Sensor LDR	47
Figura 17 - Módulo HC-05.....	48
Figura 18 - Módulo <i>Wi-Fi</i> ESP8266.....	49
Figura 19 - Fluxo de dados do projeto.....	60
Figura 20 - Modelo lógico do banco de dados.....	62
Figura 21 - Criação das tabelas via comando SQL	63
Figura 22 - Painel de gerenciamento	64
Figura 23 - Estrutura básica do PHP.....	64
Figura 24 - Funções de acesso a base de dados.....	65
Figura 25 - JSON utilizado no PHP	66
Figura 26 - Simulação	66
Figura 27 - Estrutura do DHT11	67
Figura 28 - Variação de resistência x luminosidade	68
Figura 29 - Estrutura sensor KY-038.....	69
Figura 30 - Resumo dos recursos do Arduino Mega2560	71
Figura 31 - Adaptador ESP8266 ESP-01	73
Figura 32 - Módulo Relé 2 Canais.....	75

Figura 33 - Estrutura básica de um programa para Arduino	76
Figura 34 - Utilização da biblioteca DHT.h	77
Figura 35 - Programação para o módulo Wi-Fi	78
Figura 36 - JSON de dados enviados pelo Arduino	80
Figura 37 - Comando para criação de projetos em Ionic.....	82
Figura 38 - Estrutura da pasta app.....	82
Figura 39 - Arquivo app-routing.module.ts	83
Figura 40 - Estrutura arquivo HTML	84
Figura 41 - Importação API HttpClient.....	85
Figura 42 - Implementação do método GET	85
Figura 43 - Menu da aplicação	87
Figura 44 - Tela de monitoramento	88
Figura 45 - Tela de histórico de coleta	88
Figura 46 - Tela de previsão do tempo.....	89
Figura 47 - Tela de configuração.....	90
Figura 48 - Tela de logs de alteração.....	91
Figura 49 - Tela de ajuda	91
Figura 50 - Tela de informações.....	92
Figura 51 - Registros de coleta na base de dados	93
Figura 52 - Acionamento dos equipamentos	93
Figura 53 - Gráfico de variação de temperatura.....	95
Figura 54 - Gráfico de variação da umidade	95
Figura 55 - Gráfico de variação de luminosidade	96
Figura 56 - Gráfico de coleta de sonoridade	96

LISTA DE TABELAS

Tabela 1 - Temperatura ideal em cada fase de criação	24
Tabela 2 - Limites de nível de pressão sonora.....	26
Tabela 3 - Distribuição do mercado de sistemas operacionais móveis em 2017	28
Tabela 4 - Modelos do microcontrolador Arduino.....	42
Tabela 5 - Equipamentos utilizados para a montagem da simulação	58

LISTA DE ABREVIATURAS E SIGLAS

ABCS	Associação Brasileira dos Criadores de Suínos
ART	Android Runtime
°C	Grau(s) Celsius
CC	Ciclo Completo
CLI	<i>Command Line Interface</i>
CPU	<i>Central Processing Unit</i>
EMBRAPA	Empresa Brasileira de Pesquisa Agropecuária
FAWC	<i>Farm Animal Welfare Council</i>
GHz	Gigahertz
GND	<i>Graduated Neutral Density</i>
HTTP	<i>HyperText Transfer Protocol</i>
ICSP	<i>In-Circuit Serial Programming</i>
IDE	<i>Integrated Development Environment</i>
IEEE	<i>Institute of Electrical and Eletronics Engineers</i>
ISM	<i>Industrial, Scientific and Medical</i>
JSON	<i>Javascript Object Notation</i>
m	Metro(s)
mA	Miliampere
Mbps	Megabit por segundo
MHz	<i>Megahertz</i>
NTC	<i>Negative Temperature Coefficient</i>
SGDB	Sistemas de Gerenciamento de Banco de Dados
SO	Sistema Operacional
SQL	<i>Structured Query Language</i>
TCP	<i>Transmission Control Protocol</i>
UDP	<i>User Datagram Protocol</i>
UPL	Unidade de Produção de Leitões
UR	Umidade Relativa
USB	<i>Universal Serial Bus</i>
UT	Unidade de Terminação
V	<i>Volt(s)</i>

VAC	<i>Voltage Alternating Current</i>
VCC	<i>Voltage Continuous Current</i>
VDC	<i>Voltage Direct Current</i>
W	<i>Watt(s)</i>
WLAN	<i>Wireless Local Area Network</i>

SUMÁRIO

1 INTRODUÇÃO	15
1.1 OBJETIVO GERAL	16
1.2 OBJETIVOS ESPECÍFICOS	17
1.3 JUSTIFICATIVA	17
1.4 ESTRUTURA DO TRABALHO	19
2 PRODUÇÃO DE SUÍNOS.....	20
2.1 SITUAÇÃO ATUAL DA SUINOCULTURA NO BRASIL	20
2.2 BEM-ESTAR ANIMAL	22
2.2.1 Manejo da temperatura	23
2.2.2 Manejo da pressão sonora	25
2.2.3 Manejo da umidade relativa do ar	26
2.2.4 Manejo da luminosidade.....	27
3 PLATAFORMAS MÓVEIS.....	28
3.1 PLATAFORMA ANDROID.....	29
3.2 PLATAFORMA IOS.....	31
4 APLICAÇÕES MÓVEIS	33
4.1 APLICAÇÕES NATIVAS	34
4.2 WEBAPPS.....	35
4.3 APLICAÇÕES HÍBRIDAS.....	35
4.3.1 FrameWork Ionic	37
5 MICROCONTROLADORES E SENSORES	40
5.1 ARDUINO.....	41
5.1.1 Arduino Mega2560	42
5.2 SENSORES DE TEMPERATURA E UMIDADE.....	43
5.3 SENSORES DE CAPTAÇÃO DE RUÍDOS	45
5.4 SENSORES DE LUMINOSIDADE	46
5.5 CONEXÃO COM DISPOSITIVOS EXTERNOS	47
5.5.1 Módulo Bluetooth	47
5.5.2 Módulo Wi-Fi.....	48
6 TRABALHOS CORRELATOS.....	50
6.1 DESIGN AND IMPLEMENTATION OF PIG FARM MONITORING SYSTEM FOR UBIQUITOUS AGRICULTURE	50

6.2 AUTOMAÇÃO RESIDENCIAL DE BAIXO CUSTO UTILIZANDO ARDUINO	51
6.3 ESTUDO DE FRAMEWORK MULTIPLATAFORMA APLICADO AO DESENVOLVIMENTO DE UM PROTÓTIPO DE APLICATIVO MOBILE HÍBRIDO PARA O CONTROLE FINANCEIRO PESSOAL	52
6.4 DESENVOLVIMENTO DE SISTEMA AUTOMATIZADO DE MONITORAMENTO DE AMBIENTES DE PRODUÇÃO ANIMAL, UTILIZANDO UMA REDE DE SENSORES SEM FIO.....	53
6.5 SISTEMA AUTOMATIZADO DE BAIXO CUSTO PARA PRODUTORES RURAIS: CONTROLE E MONITORAMENTO DO AMBIENTE TÉRMICO NA SUINOCULTURA	54
7 AUTOMATIZAÇÃO E MONITORAMENTO DE PROCESSOS RELACIONADOS AO BEM-ESTAR ANIMAL DENTRO DA CADEIA PRODUTIVA DA SUINOCULTURA POR MEIO DE MICROCONTROLADORES INTEGRADOS A UMA APLICAÇÃO MOBILE	55
7.1 METODOLOGIA.....	55
7.1.1 Ferramentas e recursos utilizados	56
7.1.7 Fluxo dos dados e modelo de comunicação	59
7.1.1 Modelagem dos dados.....	62
7.1.3 Configuração do servidor.....	63
7.1.4 Montagem da simulação.....	66
7.1.5 Desenvolvimento para o Arduino	75
7.1.6 Desenvolvimento da aplicação	81
7.1.8 Funcionamento da aplicação	87
7.1.9 Testes do Projeto	92
7.2 RESULTADOS E DISCUSSÕES	94
8 CONCLUSÃO	99
REFERÊNCIAS.....	101

1 INTRODUÇÃO

Segundo informações da Empresa Brasileira de Pesquisa Agropecuária (Embrapa) (2019), o Brasil é o quarto maior produtor e exportador de carne suína do mundo, chegando em 2018 a 3,76 milhões de toneladas produzidas e a 730 mil toneladas exportadas. Desempenho proporcionado principalmente por mudanças organizacionais e o contínuo incremento tecnológico na produção.

A suinocultura brasileira está posicionada entre as cadeias produtivas mais avançadas do mundo. O uso de alta tecnologia nas áreas de nutrição, genética, manejo e instalações, permitiu a produção de carne com elevados padrões de qualidade (SCHMIDT, 2017).

Com a implementação da tecnologia na cadeia produtiva, além de alavancar a produção, gerou-se também melhores índices de conforto ao animal, a qual segundo Coutinho et al. (2014), o bem-estar animal é o estado de harmonia entre o animal e seu ambiente, caracterizado por condições físicas e fisiológicas ideais a qualidade de vida do mesmo.

Fator este que representa também um papel crucial no nível de perdas durante a produção, fazendo com que os produtores se adaptem as mudanças nas regras de qualidade de vida animal, regulamentos e boas práticas de manejo (COSTA; COSTA, 2015).

Dentro dos padrões de bem-estar animal da suinocultura, para que se possa ter circunstâncias ideais para um bom desenvolvimento, diversas são as variáveis a serem tratadas pelos suinocultores, dentre elas: umidade relativa, temperatura, frequência sonora e luminosidade.

Os suínos são animais homeotérmicos, pois mantêm sua temperatura interna constante e trocam calor com o ambiente. Entretanto, para casos onde a boa conduta no manejo dos animais não é respeitada, o desenvolvimento e a produtividade esperada pelos mesmos diminuem. Fator gerado principalmente pelo estresse provocado por um ambiente não confortável aos mesmos, com isso gerando inquietação e uma menor conversão alimentar, acarretando grandes prejuízos aos produtores (FEDERIZZI, 2017).

A fim de manter níveis climáticos favoráveis ao bom desenvolvimento dos animais, atualmente os produtores dispõem-se de equipamentos como ventiladores, umidificadores e desumidificadores em suas granjas. Porém, como se trata de um

procedimento manual, requer mão de obra qualificada e vigilância constante, ao qual muitas vezes geram transtornos e conseqüentemente afetam a qualidade e a produtividade dos suínos.

Com o intuito de proporcionar melhorias neste processo, algumas tecnologias estão disponíveis no mercado, dentre elas, pode ser citada a implementação de microcontroladores e sensores, a fim de possibilitar o controle automático dos equipamentos citados anteriormente, desta forma, reduzindo a margem de erros operacionais e garantindo boas condições de vida aos animais.

Além da implementação de microcontroladores e dos variados sensores para as situações que se deseja monitorar, torna-se indispensável por parte dos produtores, formas de visualizar os dados coletados de maneira ágil e prática, para o auxílio do manejo e o gerenciamento dos locais de produção.

Com base nisso, uma boa opção a ser utilizada são os dispositivos móveis, visto que, cada vez mais estes aparelhos estão no cotidiano das pessoas, devido principalmente à sua ótima portabilidade e usabilidade. Com base nisso, torna-se possível o desenvolvimento de uma aplicação móvel, para a visualização dos dados obtidos pelos microcontroladores e sensores.

No referente trabalho, propõe-se a utilização de microcontroladores no desenvolvimento de um protótipo, para o monitoramento e automatização de processos relacionados ao bem-estar animal, na cadeia produtiva da suinocultura (temperatura, umidade relativa do ar, frequência sonora e luminosidade), integrando as informações coletadas através de uma aplicação mobile, a fim de facilitar a visualização dos dados pelos suinocultores, com isso, agilizando a tomada de decisões e auxiliando no manejo, gerando assim uma maior produtividade e melhores índices de bem-estar animal.

1.1 OBJETIVO GERAL

Monitorar processos relacionados ao bem-estar animal dentro da cadeia produtiva da suinocultura através de microcontroladores, e incorporando as informações coletadas por meio de uma aplicação mobile.

1.2 OBJETIVOS ESPECÍFICOS

Os objetivos específicos deste trabalho estão listados a seguir:

- a) compreender os conceitos e processos envolvidos na cadeia produtiva de suínos;
- b) aplicar o uso de microcontroladores a fim de monitorar as variáveis de temperatura, umidade, frequência sonora e luminosidade;
- c) elencar as funcionalidades do *framework* Ionic a fim de implementar na aplicação mobile;
- d) desenvolver uma aplicação mobile a fim de dispor de um meio de monitoramento das informações obtidas pelos microcontroladores;
- e) testar o protótipo através da simulação e aplicação a serem criados.

1.3 JUSTIFICATIVA

Segundo Embrapa (2017), a tecnologia está sendo um dos principais fatores responsáveis pelo aumento exponencial da produtividade no agronegócio brasileiro, independente da escala de produção, e não deve ser confundida como algo fora da realidade. Visto que, além do forte uso dela no setor agropecuário, ela também pode ser encontrada com facilidade em diversos outros setores da sociedade.

No setor da suinocultura, ramo situado dentro da cadeia produtiva do agronegócio brasileiro, segundo Silva e Jesus (2007), o uso da tecnologia da informação é bastante necessária, pela rapidez do ciclo produtivo da atividade e o volume de informações geradas e necessárias ao acompanhamento dos índices de produtividade zootécnicos.

Outro fator crucial na modernização da suinocultura é o quesito do bem-estar animal, de acordo com Coutinho et al. (2014), quando a qualidade de vida do suíno é comprometida, fatores como o retardo ou diminuição do ganho de peso, atraso no início da reprodução, ou até mesmo em alguns casos à própria morte, podem ocorrer. Fatores estes, que com a implementação da tecnologia e a automatização de alguns processos, podem garantir uma melhora considerável na qualidade e integridade do ciclo de vida do animal.

Ainda segundo Coutinho et al. (2014), a população passou a desejar comer carne oriunda de animais que sejam criados, tratados e abatidos em sistemas que

promovam bem-estar e que trabalham em condições ambientalmente corretas. Com isso, impulsionando a aquisição de novas práticas e métodos por parte dos produtores a fim de realizar um manejo de melhor qualidade.

Com base neste cenário, observa-se a necessidade e a possibilidade de implementar meios tecnológicos a fim de monitorar e auxiliar o manejo dos produtores no quesito de bem-estar animal. Desta forma, mantendo as variáveis de frequência sonora, luminosidade, umidade relativa e temperatura em níveis consideráveis normais para uma boa qualidade de vida dos animais.

Dentre as tecnologias possíveis de aplicação neste contexto, estão os microcontroladores, como por exemplo o Arduino, juntamente com sensores específicos para cada situação que se deseja abranger. Em resumo, o Arduino é uma pequena placa de microcontrolador contendo um plugue de conexão *Universal Serial Bus* (USB) que permite a sua ligação com um computador. Além disso, há uma infinidade de outros terminais que permitem a conexão com dispositivos externos, como motores, relés, sensores, entre outros (MONK, 2013).

Diversas são as vantagens em seu uso, dentre elas: baixo custo de aquisição, ambiente de programação simples e claro, *software* e *hardware* de código aberto e extensível, multiplataforma a qual pode ser executado em Windows, Macintosh OSX e Linux (ARDUINO, 2018, tradução nossa).

A fim de possibilitar a visualização e gestão das informações coletadas, através do microcontrolador e dos sensores por parte dos produtores, é possível a utilização de dispositivos móveis, devido a sua boa usabilidade e portabilidade. Juntamente a uma aplicação mobile utilizando-se de um *framework* para seu desenvolvimento, em específico o Ionic.

Em termos gerais, *framework* é uma junção de códigos comuns entre vários projetos de softwares que tem por finalidade prover um funcionamento genérico. Segundo Santos e Carvalho (2018), a utilização desta tecnologia possibilita a obtenção de boa estrutura para implementação, disponibiliza recursos que proporcionam a facilidade para construir um projeto de software e caso o cumprimento pleno da documentação seja realizado, ajuda a diminuir custos estabelecidos e aumenta a produtividade dos desenvolvedores.

Outra vantagem com relação ao desenvolvimento com frameworks como o Ionic, é a possibilidade de se produzir aplicativos híbridos, ou seja, que possuem compatibilidade com mais de um sistema operacional, como por exemplo o Android e

o IOS. Com isso, tornando-se uma atividade mais prática e econômica, visto que um único código fonte pode ser compilado para várias plataformas (TAVARES, 2016).

Diante de todas as situações e tecnologias citadas, torna-se possível e viável a implementação de um protótipo para controle e monitoramento de variáveis relacionadas ao bem-estar animal na cadeia produtiva da suinocultura, visando aumentar a produtividade e auxiliar o manejo dos produtores.

1.4 ESTRUTURA DO TRABALHO

O trabalho em questão é formado por oito capítulos, a qual o primeiro destes, faz uma abordagem inicial referente ao tema proposto, seguido pelo objetivo geral, objetivos específicos, justificativa e sua estrutura.

O segundo capítulo apresenta as informações referentes a produção de suínos, com a situação atual em que se encontra no Brasil e detalhes do manejo das variáveis que se pretende abordar com a implementação dos sensores.

O terceiro capítulo descreve a respeito das plataformas móveis, apresentando informações e características das mais bem posicionadas no mercado atual.

No quarto capítulo é realizada uma abordagem referente as aplicações móveis, relatando as principais metodologias de desenvolvimento e dando destaque aos frameworks multiplataformas.

O quinto capítulo descreve informações relacionadas aos microcontroladores e sensores, expondo alguns exemplos disponíveis no mercado, passíveis de serem utilizados no trabalho proposto.

No sexto capítulo é descrito os trabalhos correlatos ao objeto de estudo do referente trabalho.

O sétimo capítulo, refere-se a metodologia do trabalho proposto, onde é descrito detalhadamente todos os procedimentos realizados na confecção do mesmo. Também neste, é contido os resultados obtidos com a projeto.

No oitavo capítulo, aborda-se a conclusão alcançada com a pesquisa e desenvolvimento do projeto, bem como as sugestões de trabalhos futuros.

Por fim, encontram-se as referências bibliográficas utilizadas no desenvolvimento do trabalho.

2 PRODUÇÃO DE SUÍNOS

A produção de suínos no Brasil, é caracterizada principalmente pelo formato de confinamento, que pode compreender todos os ciclos de vida dos animais, que variam desde o nascimento até o abate. Método este, que visa principalmente a produção em larga escala, focando ao máximo em técnicas para que se tenha a possibilidade de extrair o maior rendimento e produtividade possíveis dos animais (GONÇALVES, 2012).

Dentre os principais sistemas de produção de suínos encontrados no território brasileiro, podem ser citados o Ciclo Completo (CC), Unidade de Produção de Leitões (UPL) e a Unidade de Terminação (UT) (EMBRAPA, 2003).

De acordo com Castro (2016), esses sistemas de produção possuem as seguintes características e objetivos:

- a) ciclo completo: sistema pioneiro de criação de suínos, ao qual envolve todo o ciclo de vida do animal, desde a reprodução até o leitão possuir o peso de abate, na faixa de 100 kg;
- b) unidade produtora de leitões: é responsável pela reprodução e criação de leitões, em média até os mesmos atingirem um peso de 27 kg o que dura cerca de 65 dias, após isso é realizada a venda;
- c) unidade de terminação: encarregada pela fase de crescimento dos animais, ao qual adquirem os leitões produzidos pela UPL na faixa de 27 kg de peso corporal e os criam até atingirem uma média de 100 kg, após isso encaminhando os mesmos para o abate.

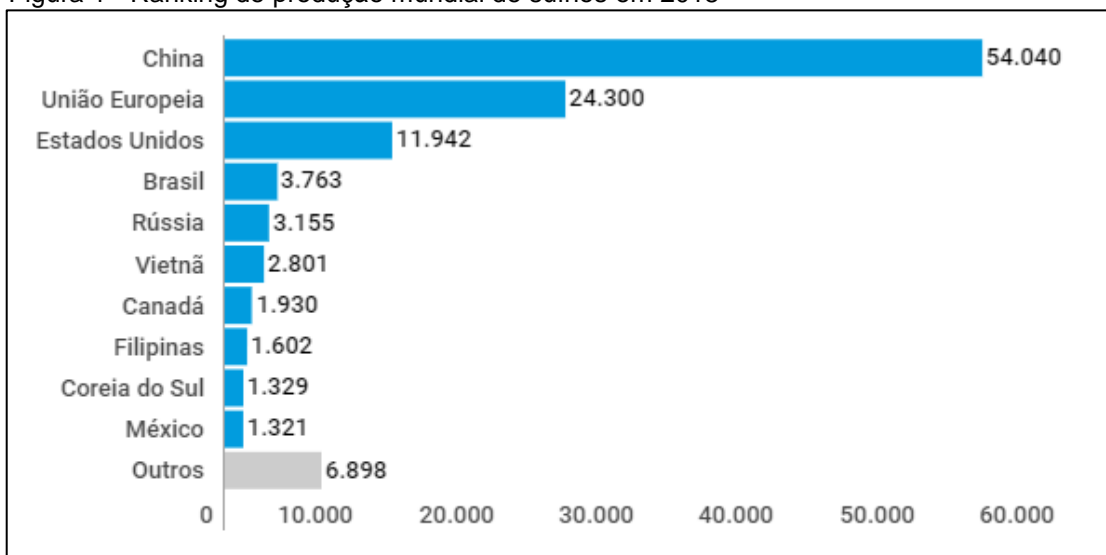
A produção independente do sistema de criação em que ela está contida, pode ser denominada como um dos fatores principais para que se possa alcançar uma boa qualidade na carne dos animais, visto que, é dentro da granja que as boas práticas de segurança e sanidade são aplicadas, garantindo com isso um alimento de qualidade na mesa do consumidor.

2.1 SITUAÇÃO ATUAL DA SUINOCULTURA NO BRASIL

A cadeia produtiva da suinocultura brasileira está posicionada entre as principais potências mundiais no *ranking* de produção. De acordo com a Embrapa (2019), o Brasil está situado em quarto lugar na classificação mundial dos produtores

de suínos, chegando à marca de 3,7 milhões de toneladas produzidas em 2018 e com aumento de 5,75%% em relação ao ano de 2017. A figura 1 ilustra o *ranking* mundial de produção em toneladas.

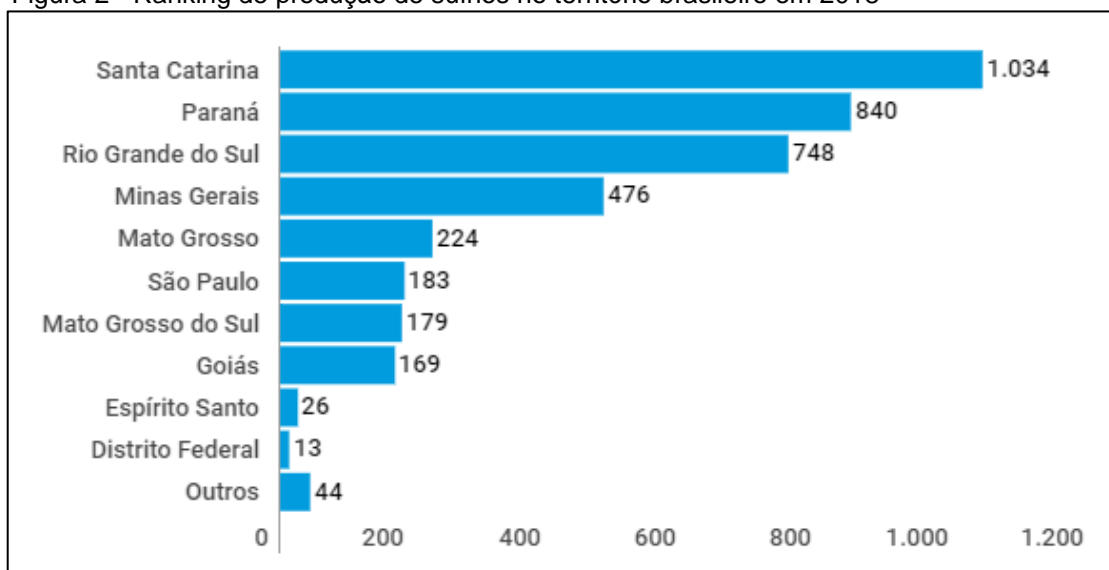
Figura 1 - Ranking de produção mundial de suínos em 2018



Fonte: Embrapa (2019).

Em âmbito de território nacional, o estado de Santa Catarina lidera o *ranking* de produção de suínos no Brasil, superando a marca de 1 milhão de toneladas produzidas no ano de 2018 (EMBRAPA, 2019). A figura 2 ilustra o gráfico com o *ranking* de produção por estado no território brasileiro.

Figura 2 - Ranking de produção de suínos no território brasileiro em 2018



Fonte: Embrapa (2019).

Diversos são os fatores que influenciam para o Brasil atingir esse número expressivo, a qual segundo Schmidt (2017), o uso da alta tecnologia nas áreas da genética, nutrição, instalações e manejo, permitiu que a produção de carne chegasse a elevados padrões de qualidade, com isso gerando maior consumo e credibilidade ao setor.

Apesar da boa colocação do Brasil no mercado mundial de suínos, fazer com que esta sustentabilidade se mantenha em alta, não é uma tarefa simples de ser realizada, segundo Schmidt (2017) alguns pontos precisam ser priorizados para manter-se nesta constante crescente, dentre eles a biossegurança, sanidade e a contínua implementação da tecnologia no manejo e cuidado dos animais, são algumas das principais medidas a serem verificadas.

Outro fator de suma importância para o crescimento do mercado de suínos, está no quesito do bem-estar animal, de acordo com Cecchin (2016), nos últimos anos a exigência dos consumidores em todo o mundo, com relação à qualidade dos produtos alimentícios, vem aumentando consideravelmente. Fatores como a higiene e qualidade de vida dos trabalhadores e animais envolvidos no sistema de produção, estão cada vez mais em pauta e ganhando importância na mesa do consumidor final. Com isso, os países com melhor estrutura nos processos de produção, se sobressaem sobre os demais.

2.2 BEM-ESTAR ANIMAL

Com base em todos os elementos existentes a serem analisados para definição do estado de bem-estar, pode-se ter um desfecho considerável extenso. Porém, em termos gerais segundo Coutinho et al. (2014), o bem-estar animal é o estado de harmonia entre o animal e seu ambiente, caracterizado por condições físicas e fisiológicas ideais ao seu bom desenvolvimento.

O grau de competência no nível de conforto animal, está diretamente atrelado a exploração bem planejada e manejada por parte dos produtores. O manejo dos animais e a maneira como estes são tratados, influenciam diretamente na sua produtividade e consequentemente no seu rendimento final (CECCHIN, 2016).

De acordo com a FAWC (2009, tradução nossa), existem cinco liberdades para que os animais mantidos presos em confinamento pelo homem, necessitam para que tenham uma boa qualidade de vida, que são:

- a) liberdade de fome e sede: promovidos pelo acesso a água fresca e comida adequada;
- b) liberdade de desconforto: proporcionado por um ambiente com boas condições de habitação;
- c) liberdade da dor, lesão ou doença: possibilitado pela prevenção ou diagnóstico eficaz no tratamento de doenças;
- d) liberdade de expressão do comportamento normal: fornecidos por instalações adequadas, com espaço adequado e companhia de sua espécie animal;
- e) liberdade de medo e angústia: promovida por tratamentos específicos que evitam o sofrimento mental.

Entre os principais fatores que influenciaram recentemente a alta na relevância do bem-estar animal na produção de suínos, está no fato de que o consumidor final, têm passando por um processo de desenvolvimento educacional e econômico, com isso, questionando-se a respeito de quais circunstâncias os alimentos que estão sobre sua mesa estão sendo produzidos (COSTA; COSTA, 2015). Dessa forma, influenciando diretamente as condições de produção, transporte e abate dos animais, visto que os produtores se sentem pressionados pelo mercado.

Neste contexto, a verificação do bem-estar animal na suinocultura, pode ser vista em diversas situações, principalmente relacionadas ao manejo e ao ambiente de produção, podendo ser verificado por exemplo nas variáveis de umidade relativa, temperatura, frequência sonora e luminosidade em que os animais estão sendo submetidos nos locais de produção (COSTA; COSTA, 2015).

2.2.1 Manejo da temperatura

Os suínos são animais homeotérmicos, ou seja, possuem a capacidade de manter sua temperatura interna constante e trocam calor com o ambiente. Entretanto, para que esses animais não sofram nenhum tipo de alteração comportamental e/(ou) fisiológica, o ambiente em que os mesmos estão situados, deve estar em um nível de conforto térmico adequado (CECCHIN, 2016).

A criação de suínos no território brasileiro, possui uma série de desafios relacionados ao clima, que podem influenciar diretamente na produtividade do setor. Fato este, acarretado principalmente pelas altas temperaturas proporcionadas pelo

clima tropical. Com isso, fazendo com que o sistema de termorregulação pouco eficaz dos suínos, gere dificuldade em manter sua temperatura constante, provocando diminuição no desempenho produtivo e redução da resistência imunológica (SOUZA, 2014).

Já de acordo com os estudos de Kiefer et al. (2010), suínos submetidos a estresse por calor, permanecem mais tempo deitados, ou seja, menos tempo se alimentando e mais tempo dormindo. Consequentemente, o consumo de ração, ganho de peso e o índice de conversão alimentar, tem uma redução significativa.

Para os animais submetidos a temperaturas adversas da faixa ideal, apresentam carcaças com peso inferior, com maior rendimento e menor percentual de carne, Kiefer et al. (2010). Como resultado disso, os índices gerados de bonificação aos produtores no momento do abate, são consideravelmente menores, influenciando com isso, uma queda significativa no faturamento obtido.

Outra adversidade ocasionada pela irregularidade da temperatura para os suínos, de acordo com Souza (2014), está relacionada ao fato, de que os animais podem começar a adquirir fatores de risco às doenças respiratórias, como por exemplo rinite e pneumonia. Devido a cronicidade destas doenças, nem sempre são percebidas pelos produtores, causando prejuízos significativos.

A criação de suínos requer para cada estágio de produção dos animais, determinadas faixas de temperatura (ROHR; COSTA; COSTA, 2016) como pode ser observado na tabela 1.

Tabela 1 - Temperatura ideal em cada fase de criação

Fase	Temperatura ideal (°C)
Reprodução (fêmeas e machos)	18 a 25
Maternidade – porca	16 a 21
Maternidade – leitão	34 a 30 (decrecente ao longo da fase)
Creche	30 a 23 (decrecente ao longo da fase)
Terminação	18 a 23

Fonte: Rohr, Costa e Costa (2016).

Segundo a Associação Brasileira dos Criadores de Suínos (ABCS) (2016), os produtores não devem tomar apenas o cuidado em manter a ambiência conforme ilustrado na tabela 1, mas também, minimizar a amplitude térmica das granjas, ou

seja, a variação de temperatura ao longo do dia. Nas condições climáticas presentes no território brasileiro, isso só se torna possível com o auxílio de equipamentos específicos para a climatização de ambientes, como por exemplo: ventiladores, aspersores e exaustores.

A temperatura no interior das instalações, tem importância crucial para o triunfo da suinocultura, de acordo com Souza (2014), apesar de existirem diversos estudos realizados referente às condições do ambiente térmico das granjas, e com muitos deles possuindo referências e valores de clima diferentes para cada fase de vida dos animais, o importante para os produtores é possuir uma base ao qual podem seguir e obter conhecimento dos empecilhos que podem adquirir em caso de não realizar as medidas necessárias.

2.2.2 Manejo da pressão sonora

O ruído pode ser definido como uma vibração emitida no ar por um objeto que está vibrando, se caracterizando pela existência de amplitudes e frequências sonoras não harmônicas ocorrendo ao mesmo tempo. Segundo a ABCS (2014), o ar transmite essas vibrações ao tímpano, ao qual vibra como reação às vibrações da pressão.

Em um ambiente de produção de suínos, de acordo com Cecchin (2016), a pressão sonora está relacionada principalmente aos sons emitidos pelos animais e equipamentos. Com isso, é de suma importância a realização de avaliações constantes do nível de ruído do ambiente de produção, visto que são diversos os fatores problemáticos em que um nível de sonoridade elevado pode ocasionar, dentre eles: alterações nos níveis de ansiedade, sono, e atenção, como também proporcionando mudanças no comportamento do sistema nervoso, fadiga mental e irritabilidade.

De acordo com Souza (2014), outra característica importante para a verificação do nível de vocalização dos animais, está no fato de que pode ser uma ótima forma de avaliar as condições de bem-estar animal. Devido ao fato de que a vocalização emitida pelos suínos, pode representar o estado em que o animal se encontra, seja ele gerado por motivos espontâneos ou por resultado de um evento externo.

Ainda de acordo com os estudos de Souza (2014), os danos causados pela exposição a um dado ruído, depende diretamente do nível e de sua duração em que se é submetido. A tabela 2 representa a relação dos níveis máximos de pressão sonora permitidos e a duração de tempo para cada situação.

Tabela 2 - Limites de nível de pressão sonora

Nível da pressão sonora dB (A)	Tempo de máxima exposição diária permissível
85	8 horas
90	4 horas
95	2 horas
100	1 horas
105, 110 e 115	30, 15 e 7 minutos respectivamente

Fonte: Souza (2014).

2.2.3 Manejo da umidade relativa do ar

A Umidade Relativa (UR) do ar pode ser entendida como a quantidade de vapor da água existente no ar em uma determinada temperatura, sendo possível expressá-la em porcentagem. Com isso, a UR está diretamente atrelada a temperatura do ambiente, visto que, quanto mais quente estiver o clima, maior será a capacidade de retenção de vapor da água pelo ar e vice-versa (PINTO, 2012).

Na suinocultura, a UR pode causar influência direta no nível bem-estar animal, já que a mesma pode alterar o ambiente de termoneutralidade das granjas, com isso, afetando a dissipação de calor por perda evaporativa de água dos suínos. Desta forma, gerando inquietação e uma perda significativa no consumo alimentar dos animais (MELLAGI et al., 2010).

De acordo com Bortolozzo et al. (2010), a faixa ideal da umidade relativa do ar para o conforto térmico dos suínos, está entre 60 e 80%. Além disso, esses animais não contam com um mecanismo de proteção contra altas temperaturas muito sofisticado, com isso usando apenas a mudança de comportamento e a sua própria ofegação para controlar o mal-estar proporcionado por níveis de UR alterados. Desse modo, gerando um grande consumo energia e desgaste físico nos animais.

2.2.4 Manejo da luminosidade

A luminosidade, em termos gerais representa a presença de luz em um determinado ambiente. Na suinocultura, é de crucial importância realizar o manejo adequado referente a quantidade de luz em que os animais são expostos. De acordo com a ABCS (2014), os suínos com idade adulta preferem áreas menos iluminadas para repousar em seu período de descanso e já para os leitões, os locais escuros podem gerar medo e desconforto, influenciando com isso o seu nível de bem-estar.

Com relação aos problemas acarretados pela incidência direta da luz solar nos animais, estão relacionados principalmente a queimaduras com consequente desconforto, dor ou aborto em matrizes em fase de gestação, a qual nesta última situação, podendo gerar sérios prejuízos aos produtores (ROHR; COSTA; COSTA, 2016).

3 PLATAFORMAS MÓVEIS

O mercado de *smartphones* ou em uma tradução literal “telefones inteligentes”, no Brasil e no mundo, vem crescendo a um ritmo extremamente elevado nos últimos anos. Em agosto de 2018, o Brasil atingiu a marca de 234,7 milhões de telefones celulares, chegando a uma densidade de 111,96 celulares a cada 100 habitantes, ou seja, o número de celulares já superou a quantidade de habitantes (TELECO, 2018).

Alguns dos principais fatores que influenciaram nestes expressivos números, está na questão da intensa implementação tecnológica na busca de novas descobertas e constantes melhorias, principalmente voltadas ao hardware e sistemas operacionais em que esses dispositivos são baseados. Com isso, fazendo com que os consumidores sempre almejem manter-se atualizados ao mercado.

As plataformas ou sistemas operacionais móveis, foram criadas com intuito de disponibilizar uma melhor interação entre os usuários e os *smartphones*, tendo como responsabilidade manipular todos os recursos do dispositivo, como: câmera, GPS, lista telefônica, entre outros (MAZUCHETTI, 2015).

Atualmente é possível encontrar uma gama diversificada de sistemas operacionais para dispositivos móveis, porém dentre elas destacam-se o Android da Google e o IOS da Apple, que de acordo com pesquisa disponibilizada (tabela 3) pela Gartner (2018) em 2017 essas duas plataformas juntas chegaram a 99,9% de todo o mercado mundial de *smartphones*.

Tabela 3 - Distribuição do mercado de sistemas operacionais móveis em 2017

Operating System	2017 Units	2017 Market Share (%)
Android	1,320,118.1	85.9
IOS	214,924.4	14.0
Other OS	1,493.0	0.1
Total	1,536,535.5	100.0

Fonte: Gartner (2018).

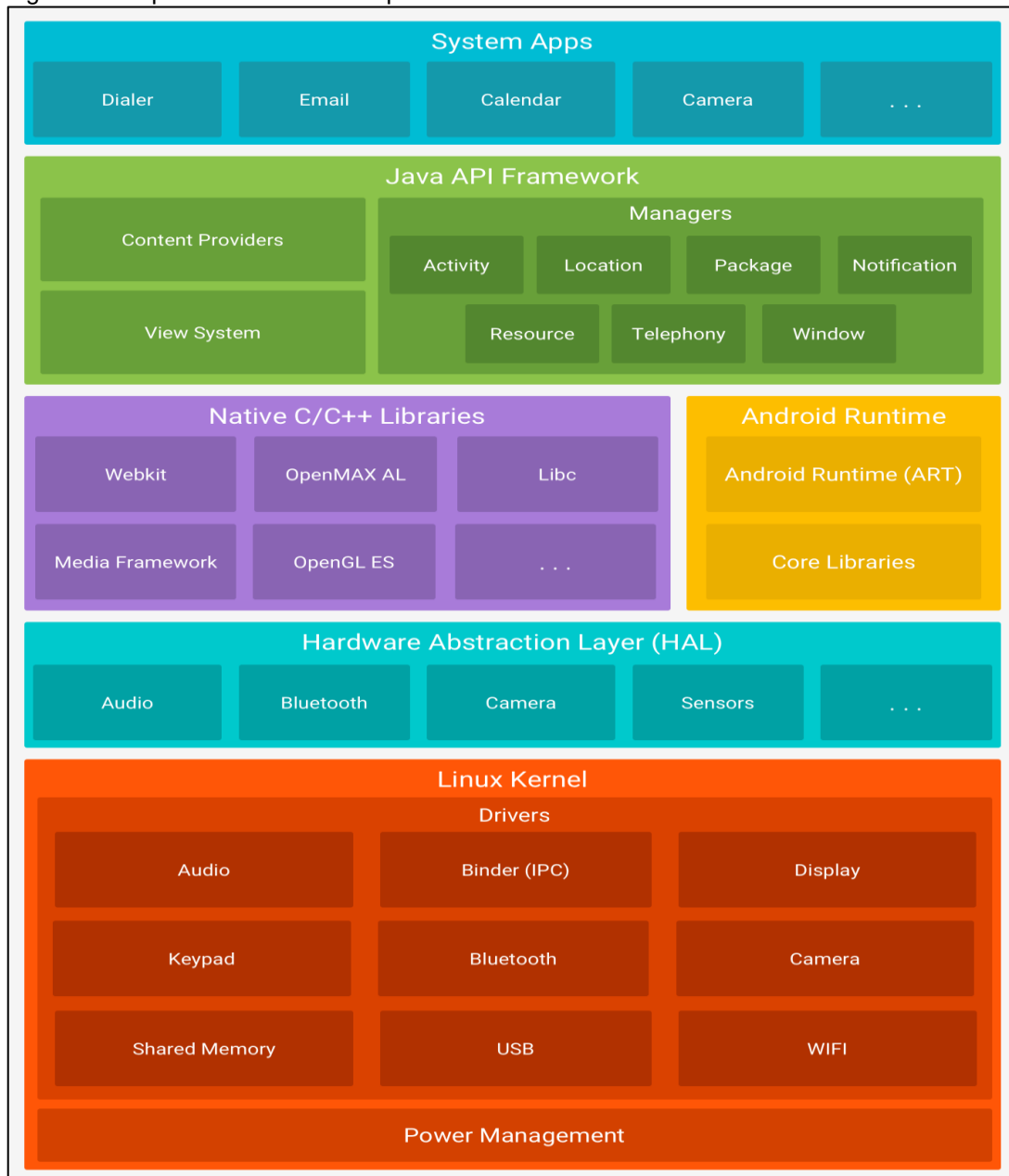
3.1 PLATAFORMA ANDROID

É uma plataforma para dispositivos móveis baseada no núcleo do operacional Linux. Inicialmente foi desenvolvida por um startup de mesmo nome, a Android Inc., ao qual em 2005, foi adquirida pela Google (LEE, 2011, tradução nossa). A versão inicial desta plataforma foi lançada em 2008, como nome de Android Alpha 1.0 e desde então não parou mais de evoluir, chegando até a versão atual, o Android Pie 9.0.

O Android é a primeira linguagem de programação para dispositivos móveis livre e de código aberto (*open source*), o que se torna um ponto crucial para a sua constante evolução, visto que diversas empresas e desenvolvedores podem contribuir para a melhora contínua da plataforma. Fato este, que se torna muito atrativo também aos fabricantes de celulares, já que os mesmos podem usar esta plataforma em seus aparelhos sem pagar nada por isso, como também, podendo realizar suas melhorias ou alterações específicas sem precisar compartilhar com os demais concorrentes (LECHETA, 2015).

De acordo com o site oficial da plataforma Android (2019), a arquitetura do sistema é composta por seis grandes camadas, conforme pode ser visto na figura 3.

Figura 3 - Arquitetura do sistema operacional Android



Fonte: Android (2019).

Conforme ilustrado na figura 3, as seis camadas do sistema operacional possuem os seguintes objetivos e funcionalidades (ANDROID, 2019):

- aplicativos do sistema: nesta camada superior, pode-se encontrar um conjunto de aplicativos que vem instalados junto a plataforma, dentre eles, os de e-mail, navegador de internet, calendário, entre outros;
- Java API *framework*: composta pelo conjunto completo de recursos do sistema operacional programados na linguagem Java. São formadas

- basicamente por blocos de programação, que podem ser reutilizados na criação de aplicativos para o acesso a variados recursos do dispositivo;
- c) bibliotecas C/C++ nativas: diversos são os componentes e serviços do sistema operacional Android, que são desenvolvidos através de código nativo, ao qual reivindicam bibliotecas nativas programadas em C e C++. O Android disponibiliza as Java Framework API a fim de possibilitar o acesso de aplicativos a estas bibliotecas;
 - d) Android runtime: para as versões do Android superiores a 5.0, os aplicativos funcionam de tal forma que executam seus processos com uma instância própria do Android Runtime (ART). Onde este possui por finalidade executar várias máquinas virtuais em dispositivos de baixa memória. Dentre os recursos fornecidos pelo ART estão a coleta de lixo otimizada e uma boa compatibilidade para a depuração dos aplicativos;
 - e) camada de abstração de hardware: responsável por fornecer interfaces que disponibilizam a capacidade do hardware do dispositivo para a estrutura do Java API. A estrutura que esta camada é formada, é composta basicamente por módulos de bibliotecas, que implementam uma interface para determinados componentes de hardware, como por exemplo a câmera e o *Bluetooth*;
 - f) núcleo Linux: esta camada é responsável por armazenar todos os drivers de baixo nível utilizados pelos componentes de hardware do dispositivo.

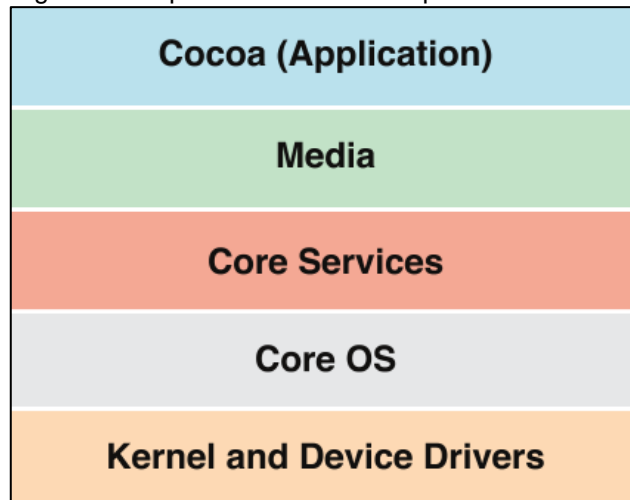
3.2 PLATAFORMA IOS

É um sistema operacional desenvolvido pela Apple, teve seu lançamento no ano de 2007 com a versão 1.0 (iPhone OS), ao qual em 2010 teve sua nomenclatura modificada para iOS, devido ao fato de que outros dispositivos além do iPhone, como iPod's e iPad's, passaram também a utilizá-lo (PILAR, 2013).

De acordo com Lecheta (2017), uma das principais características da plataforma IOS, está no fato de que ela possui uma exclusiva integração com os hardwares da Apple, muito famosa por desenvolver ótimos conjuntos de hardwares, com isso garantindo um excelente desempenho de todo o sistema.

A arquitetura do sistema operacional IOS é dividida em cinco níveis (APPLE, 2015, tradução nossa), conforme ilustrado na figura 4.

Figura 4 - Arquitetura do sistema operacional IOS



Fonte: Apple (2015).

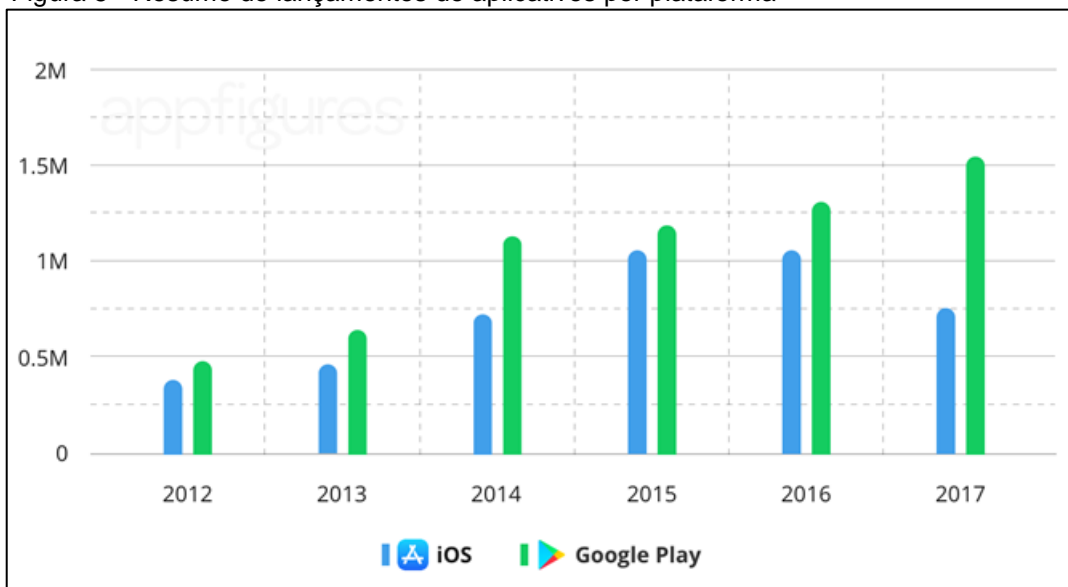
Conforme ilustrado na figura 4, os níveis deste sistema operacional (SO), possuem as seguintes finalidades e aplicabilidades (APPLE, 2015, tradução nossa):

- a) *cocoa (application)*: inclui tecnologias para criar a interface do usuário com as aplicações, onde possui como responsabilidade responder aos eventos realizados pelos usuários e gerenciar o comportamento dos aplicativos;
- b) *media*: engloba as tecnologias especializadas para reproduzir, gravar e editar mídias audiovisuais e para renderizar e animar gráficos 2D e 3D;
- c) *core services*: contém muitos serviços e tecnologias fundamentais que vão desde a contagem de referência automática e comunicação de rede de baixo nível até a manipulação de strings e formatação de dados;
- d) *core OS*: define as interfaces de programação relacionadas ao hardware e rede, incluindo interfaces para executar tarefas de computação de alto desempenho na CPU e na GPU de um dispositivo;
- e) *kernel and device drivers*: consiste nos drivers do dispositivo e outros componentes de baixo nível. Esta camada inclui suporte para sistemas de arquivos, rede, segurança, comunicação entre processos, linguagens de programação, drivers do dispositivo e extensões para o kernel.

4 APLICAÇÕES MÓVEIS

Da mesma forma que as plataformas móveis, citadas no capítulo anterior, tiveram nos últimos anos um intenso crescimento, as aplicações móveis não ficaram para trás, ao qual só no ano de 2017, foram lançados em torno de 1.5 milhões de novos aplicativos na Google Play (Android) e 755 mil da App Store (IOS) (APPFIGURES, 2018, tradução nossa). A figura 5 demonstra o resumo do desenvolvimento de aplicativos dos anos de 2012 à 2017 voltados para estas plataformas.

Figura 5 - Resumo de lançamentos de aplicativos por plataforma



Fonte: Appfigures (2018).

Frente a este expressivo crescimento, os desenvolvedores e empresas buscam cada vez mais alternativas para melhorar a performance de desenvolvimento e redução de manutenções de suas aplicações. Sendo que, são diversas as formas e metodologias possíveis em que é possível basear-se para construir aplicações, com isso tornando-se uma tarefa árdua, disponibilizar aplicativos que funcionem de forma integrada em todas as plataformas e tecnologias disponíveis atualmente (DEVMEDIA, 2015).

Com relação ao mercado de aplicações móveis, está concentrado principalmente em três modalidades, que são: nativa, *webapps* ou também conhecidas por *mobile web sites* e as híbridas (MAZUCHETTI, 2015). Nos próximos subcapítulos serão descritos as principais características e objetivos de cada uma.

De acordo Nascimento e Bonetti (2016), cada uma das categorias de desenvolvimento *mobile*, possuem características e objetivos distintos umas das outras, sendo essencial realizar uma análise previamente ao desenvolvimento, a fim de identificar qual a melhor escolha de acordo com cada situação e necessidades. A figura 6 ilustra um comparativo entre as modalidades de aplicações móveis, demonstrando os principais recursos atendidos por cada uma delas.

Figura 6 - Comparativo entre *native apps*, *webapps* e aplicações híbridas

	Acesso ao Dispositivo	Performance	Tempo Desenvolvimento	App Store	Multi Plataforma
Nativo	Sim	Sim	Caro	Sim	Não
Web	Parcial	Sim*	Ótimo	Não	Sim
Híbrido	Sim	Sim*	Ótimo*	Sim	Sim*

* Depende de como a aplicação é desenvolvida.

Fonte: Nascimento e Bonetti (2016).

4.1 APLICAÇÕES NATIVAS

São desenvolvidas utilizando ferramentas e recursos de uma linguagem de programação específica de cada plataforma, como por exemplo o Java para o Android e o Objective-C para o IOS (NASCIMENTO; BONETTI, 2016).

Pelo fato desta categoria de desenvolvimento estar voltada diretamente a uma plataforma específica e com suas aplicações instaladas diretamente no Sistema Operacional (SO) dos dispositivos, a parte operacional das mesmas pode ser realizada de forma *off-line* (MAZUCHETTI, 2015).

Outras características desta metodologia, está no fato de que os aplicativos possuem maior desempenho, visto que foram projetados para um SO específico, como também dispõem do acesso facilitado aos recursos dos dispositivos, como:

câmera, acelerômetro, GPS, bússola entre outros, tornando os mesmos mais completos neste quesito (NASCIMENTO; BONETTI, 2016).

Apesar do desempenho e compatibilidade com os dispositivos das plataformas em que os aplicativos foram desenvolvidos serem elevados, o custo benefício de aplicações nativas pode ser considerado baixo, se relacionado ao fato de que os desenvolvedores precisam ter conhecimento das linguagens de programação exclusiva de cada SO. Perante isso, o tempo de aprendizado e desenvolvimento pode aumentar (DEVMEDIA, 2015).

4.2 WEBAPPS

Os *Webapps* ou *mobile web sites*, são aplicações desenvolvidas com tecnologias utilizadas geralmente no desenvolvimento de sites para web, como: HTML5, CSS e JavaScript, além ferramentas como Java e PHP (WILKEN, 2016, tradução nossa).

Possuem como principais vantagens, a portabilidade e compatibilidade com a maioria dos dispositivos móveis independente de sua plataforma, visto que são acessados diretamente de um navegador web, com isso não tendo a necessidade de instalação. Em sua maioria, estas aplicações são sites comuns da internet, porém projetos para se adequarem a dispositivos menores, principalmente por meio de tecnologias responsivas que se adaptam as mais variadas resoluções (WILKEN, 2016, tradução nossa).

Por outro lado, pelo fato desta metodologia de desenvolvimento ter seus princípios baseados na web, a mesma não tem capacidade de operar em modo *off-line*. Como também, tendo acesso parcial aos recursos e funcionalidades do dispositivo, e não sendo possível ser disponibilizadas nas lojas de aplicativos (NASCIMENTO; BONETTI, 2016).

4.3 APLICAÇÕES HÍBRIDAS

Aplicações híbridas ou também conhecidas por multiplataforma, possuem como principal objetivo, a capacidade de serem executadas em diversas plataformas, sem a necessidade de alteração do código fonte dos aplicativos. Sua estrutura é formada pela junção das metodologias *web apps* e nativa.

Diferentemente dos *web apps*, os aplicativos híbridos ficam instalados no dispositivo, tendo a possibilidade de operar em modo *off-line*, como também, podendo ser disponibilizados nas lojas de aplicativos, similar aos nativos. Normalmente são desenvolvidos em linguagem *web*, sendo interpretados pelo *browser* do SO em que estão contidos (PREZOTTO; BONIATI, 2014).

De acordo com Gok e Khanna (2013, tradução nossa), os aplicativos híbridos desempenham um papel fundamental na redução da lacuna existente entre os recursos dos navegadores e dos dispositivos, permitindo desta forma, que os desenvolvedores possam aproveitar o melhor dos dois mundos, ao qual geralmente utilizam-se de combinações de HTML5, CSS, JavaScript juntamente com os SDKs específicos de cada plataforma, como o Java no Android e o Objective-C para o IOS.

Com relação as vantagens que as aplicações híbridas possuem com relação as demais metodologias, segundo Gok e Khanna (2013, tradução nossa), são:

- a) redução no tempo de desenvolvimento: a criação de aplicativos geralmente é mais rápida, visto que requer apenas conhecimento em um único padrão de desenvolvimento, ao qual pelo fato de ser multiplataforma, não há necessidade de trabalhar com linguagens específicas;
- b) custo de desenvolvimento e recursos abundantes: pelo fato de que as aplicações híbridas são baseadas em linguagens *web*, o custo de formação de um desenvolvedor é menor, como também sendo mais fácil encontrar mão de obra disponível no mercado;
- c) custo de manutenção: são geralmente menores, em função de que está tudo centralizado em apenas um único código, não sendo necessário a reescrita de código em plataformas diferentes;
- d) acesso aos recursos do dispositivo: além de desfrutar de todos os recursos tradicionais da *web*, estes apps também dispõem do acesso dos benefícios de cada plataforma que estão contidos;
- e) distribuição através de lojas de aplicativos: assim como os aplicativos nativos, os híbridos também podem ser distribuídos nas lojas virtuais, a qual possibilita aos desenvolvedores uma forma de monetização já existente.

Com a popularização e o crescimento destas aplicações devido as vantagens que elas possuem, novas ferramentas começaram a ser lançadas no

mercado a fim de otimizar o desenvolvimento de aplicativos híbridos, chamadas de *frameworks*.

Um framework multiplataforma pode ser entendido como um aglomerado de bibliotecas, ferramentas e códigos fontes unificados em uma única linguagem, disponibilizando a possibilidade de suporte para todas as plataformas. (NASCIMENTO; BONETTI, 2016).

Dentre os colocados no top cinco dos mais utilizados pelos desenvolvedores no ano de 2018, segundo Excellent Webworld (2018, tradução nossa) e Websoptimization (2018, tradução nossa), está o Ionic.

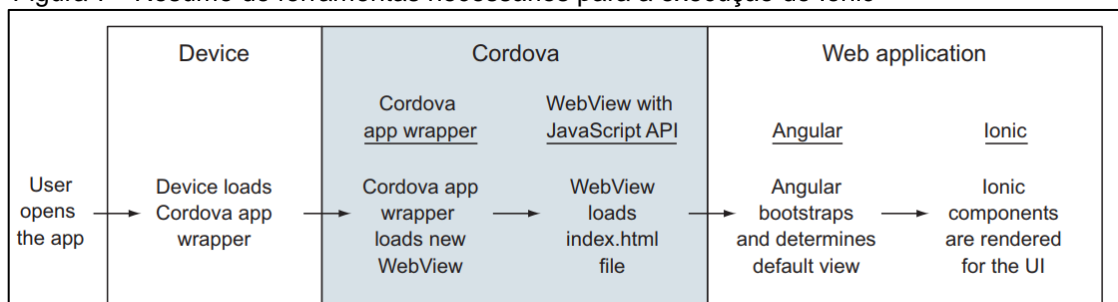
4.3.1 FrameWork Ionic

O Ionic teve como responsabilidade trazer ao mercado um conjunto de melhorias para o desenvolvimento de aplicações híbridas, ao qual anteriormente ao seu lançamento no ano de 2013, os dispositivos móveis eram relativamente lentos e somente aplicações nativas eram capazes de fornecer performance aos mesmos (WILKEN, 2016, tradução nossa).

Este *framework* pode ser entendido, como a combinação de tecnologias como HTML, CSS e JavaScript, que foram projetadas com intuito de facilitar e agilizar a construção de aplicativos híbridos com destaque visual. Contudo, para que ele possa ser executado, outras ferramentas devem acompanhá-lo, sendo elas, o Angular como framework de aplicação web JavaScript e o Cordova sendo encarregado pela construção e interpretação do aplicativo junto ao SO (MAZUCHETTI, 2015).

A figura 7 representa o resumo de como está estruturado as ferramentas necessárias para o funcionamento do Ionic.

Figura 7 - Resumo de ferramentas necessários para a execução do Ionic



Fonte: Wilken (2016, tradução nossa).

Conforme pode ser observado na figura 7, o primeiro quadro corresponde ao dispositivo, que detêm o sistema operacional. Já o segundo retrata o Cordova, que tem por finalidade a interpretação entre o SO e o aplicativo, criando uma aplicação nativa que é composta por um WebView, que possui características de uma página web, onde a aplicação é internamente executada (WILKEN, 2016, tradução nossa).

Uma outra característica marcante do Cordova é que o faz ter uma essencial utilização junto ao Ionic, é o fato do mesmo possibilitar o desenvolvimento de aplicativos móveis, sem ter qualquer necessidade de programação de códigos nativos, como Java para o Android e Objective-C para o IOS, com isso possibilitando o desenvolvimento inteiramente utilizando-se de tecnologias web (MAZUCHETTI, 2015).

Já o último estágio é representando pelo Angular e o próprio Ionic, que tem como objetivos controlar a lógicas e os dados das aplicações.

De acordo com Wilken (2016, tradução nossa), existem uma série de vantagens e recursos que o Ionic pode oferecer aos desenvolvedores, dentre elas:

- a) desenvolver aplicativos híbridos com linguagens características da web: resulta na possibilidade de criar aplicações móveis que se comportam como nativas com tecnologias, como HTML, CSS e JavaScript;
- b) desenvolvimento em Angular: Ionic possibilita estender o Angular ao ambiente móvel;
- c) uso de tecnologias modernas: este *framework* possibilita o uso de CSS3 com as mais variadas animações, suportada geralmente pela maioria dos navegadores móveis;
- d) ferramentas para o desenvolvimento: o Ionic disponibiliza opções de arrasta e solta, como também a possibilidade de geração de versões betas dos aplicativos, a fim de disponibilizar para os usuários finais com intuito de verificar a aceitação do projeto;
- e) poderosa ferramenta CLI: possibilita o gerenciamento da aplicação a ser desenvolvida, através de emuladores ou até mesmo em dispositivos conectados, facilitando com isso a pré-visualização do projeto;
- f) designer flexível: projetado para implementar diretrizes de estilo nativas, porém permitindo também, fáceis modificações das características visuais dos aplicativos.

Outra vantagem considerável deste framework, que não atrai apenas os desenvolvedores, mas também grandes empresas, está no fato de que é de código aberto, possuindo um licenciamento permissivo para a sua utilização, com isso não implicando em questões de direitos autorais para as aplicações pessoais ou comerciais desenvolvidas a partir dele (IONIC, 2019, tradução nossa).

A versão atual disponível para utilização do Ionic é a quatro, ao qual teve seu lançamento em janeiro de 2019. Uma das principais mudanças que ocorreram com relação as versões anteriores, está no fato de não possuir mais uma única e exclusiva a integração com o Angular, como por exemplo no esquema ilustrado na figura 7, mas também agora, a outras estruturas *front-end* famosas como o Vue e o React, com isso, dando uma maior liberdade de escolha aos desenvolvedores (IONIC, 2019, tradução nossa).

Outra característica marcante da versão quatro do Ionic, está na possibilidade de utilização de um único código fonte, para a execução e utilização em várias plataformas, como IOS na versão 10+, Android na versão 4.4+, desktop e WEB com *progressive web app*, ao qual a maioria dos navegadores atuais como Chrome, Edge, Firefox e Safari já provem suporte ao Ionic.

A popularidade do Ionic no mundo dos desenvolvedores já é bastante considerável e tende a cada vez mais crescer, principalmente pelo fato de disponibilizar aplicações híbridas com mesma performance de nativas e com design superior, caracterizado por aplicativos dinâmicos e responsivos que se adaptam nos mais variados tipos de telas (WILKEN, 2016, tradução nossa).

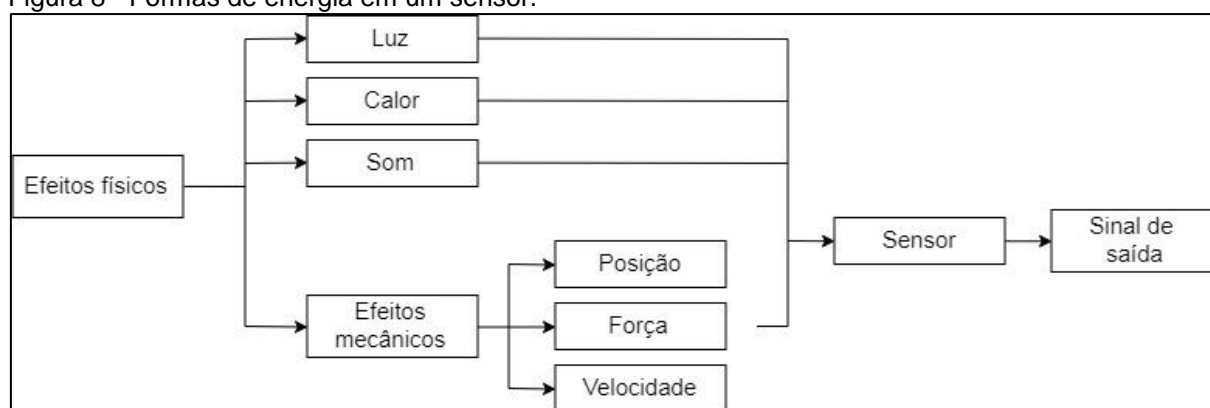
5 MICROCONTROLADORES E SENSORES

Microcontroladores podem ser entendidos como a junção de um microprocessador com interfaces de entrada e saída e outros periféricos, em apenas um único componente (POUSO, 2012).

O microprocessador, ou também conhecido por Unidade Central de Processamento, do inglês *Central Processing Unit* (CPU), tem como objetivo principal processar dados, buscando instruções na memória, decodificando e as executando. Já as interfaces de entrada e saída, disponibilizam a possibilidade de conexão com dispositivos externos como sensores, relés, entre outros (POUSO, 2012).

Já os sensores são dispositivos que de alguma forma são sensíveis aos tipos de energia presentes na natureza, como por exemplo a cinética, luminosa, térmica, e outras, e através delas relacionam as informações sobre uma grandeza previamente medida, como: aceleração, corrente, pressão, temperatura, umidade, velocidade, entre outras (THOMAZINI; ALBUQUERQUE, 2005). A figura 8 ilustra as formas de energia em um sensor.

Figura 8 - Formas de energia em um sensor.



Fonte: Thomazini e Albuquerque (2005).

De acordo com Thomazini e Albuquerque (2005), um sensor nem sempre tem as características elétricas necessárias para ser utilizado em um sistema de controle. Geralmente deve ser manipulado o sinal de saída, antes da sua leitura no sistema de controle. Isso normalmente é realizado com um circuito de interface para produção de um sinal que possa ser lido pelo controlador.

A fim de realizar o controle de algumas variáveis relacionadas ao bem-estar dos animais na cadeia produtiva da suinocultura, pode ser encontrada uma

diversificada variedade de equipamentos disponíveis no mercado, dentre eles o microcontrolador Arduino, que disponibiliza a capacidade de conexão com sensores externos, como os de temperatura, umidade, sonoridade e luminosidade.

5.1 ARDUINO

É um pequeno microcontrolador que possui vários pinos de conexão, a qual permite que outros circuitos eletrônicos externos sejam conectados a ele, como por exemplo, motores, relés, sensores, alto-falantes, entre outros. Também possui uma porta *Universal Serial Bus* (USB), que o permite se conectar a um computador com intuito de realizar o controle do fluxo de dados (MONK, 2014).

De acordo com o site oficial da plataforma (Arduino, 2019, tradução nossa), as principais vantagens são:

- a) custo: as placas possuem valor de aquisição inferiores se comparados a outras plataformas;
- b) ambiente de programação simples e claro: o *software* (IDE) possui uma boa flexibilidade, a qual é de fácil aprendizado para iniciantes e compatível para usuários experientes também aproveitarem;
- c) multiplataforma: o Arduino é capaz de ser executado nos principais sistemas operacionais existentes atualmente no mercado, como o Windows, Macintosh OSX e o Linux, diferente da maioria dos sistemas de microcontroladores que se limitam ao Windows;
- d) *software open source* e extensível: o Arduino é distribuído como ferramenta *open source* (código aberto), disponível para extensão por programadores em níveis avançados, tendo também a possibilidade de ser expandida através da biblioteca C ++;
- e) *hardware open source* e extensível: possibilita que designers de circuitos experientes possam fazer suas próprias versões do módulo, estendendo-o e aprimorando-o.

As aplicabilidades do Arduino são basicamente ilimitadas, dependendo exclusivamente da criatividade e disciplina empregada pelos desenvolvedores. Dentre os principais exemplos em que o microcontrolador é utilizado, estão na automatização de casas, carros, escritórios e ambientes de produção.

O mercado atual deste microcontrolador, possui uma boa diversidade de modelos, a tabela 4 representa alguns deles com suas principais especificações.

Tabela 4 - Modelos do microcontrolador Arduino

Modelo	V _{in} (V)	V _{pin} (V)	Clock (MHz)	Digitais	Analógicas	PWM	UART	Flash (Kb)
Arduino Uno - R3	7-12	5V	16	14	6	6	1	32
Arduino Leonardo	7-12	5	16	20	12	7	1	32
Romeo V2.0	7-12	5	16	20	12	7	1	32
Julietta	7-12	5	16	14	8	6	1	32
Arduino Mega2560	7-12	5	16	54	16	15	4	256
Arduino Due	7-12	3.3	84	54	12	12	4	512
Arduino Micro	7-12	5	16	20	12	7	1	32
Arduino Pro Micro	5-12	5	16	12	4	5	1	32
Arduino Pro Mini 3.3V	3.3-12	3.3	8	14	6	6	1	32
Arduino Pro Mini 5V	5-12	5	16	14	8	6	1	32

Fonte: Adaptado de Robocore (2017).

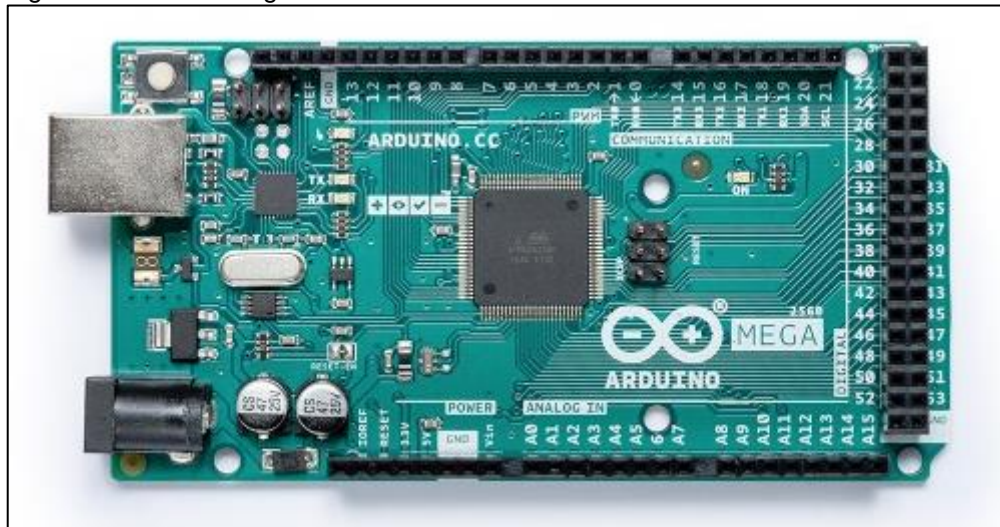
De acordo com os modelos ilustrados na tabela 4, o Arduino Mega2560 está entre os que possui maior robustez, devido a sua disponibilidade superior de conexões. No próximo subcapítulo, será relatado com mais detalhes as suas características e especificações.

5.1.1 Arduino Mega2560

O modelo Mega2560 possui seus princípios baseados no ATmega2560. Nele pode ser encontrado 54 pinos de entrada / saída digitais, ao qual, 15 destes podem ser usados como saídas PWM, 16 entradas analógicas, 4 UARTs (portas seriais de *hardware*), uma conexão USB, um oscilador de cristal de 16 MHz, conector de alimentação, um conector *In-Circuit Serial Programming* (ICSP) e um botão de reset (ARDUINO, 2019, tradução nossa).

Dentre os modelos da família de microcontroladores Arduinos, o Mega (figura 9) é um dos mais robustos, visto que está voltado principalmente para projetos mais complexos e que exigem maior número de conexões.

Figura 9 - Arduino Mega2560



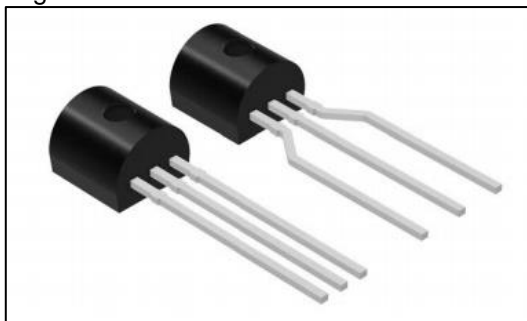
Fonte: Arduino (2019).

5.2 SENSORES DE TEMPERATURA E UMIDADE

Sensores de temperatura e umidade, são componentes que tem por finalidade realizar medições do grau de temperatura e da quantidade de água existente no ar de um determinado ambiente em que estão contidos. Dentre as variedades existentes no mercado, pode ser citado os modelos LM35 para temperatura e HR202L para umidade.

O modelo LM35 possui capacidade de operar em temperaturas que podem variar de -55°C até $+150^{\circ}\text{C}$, tendo uma precisão de 0.5°C e pode receber alimentação variando de 4V até 30V, (TEXAS INSTRUMENTS, 2017, tradução nossa), conforme pode ser observado na figura 10.

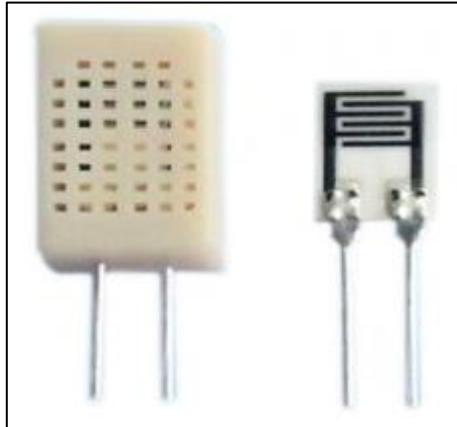
Figura 10 - Sensor LM35



Fonte: Texas Instruments (2017).

O componente HR202L (figura 11), apresenta aptidão de verificar níveis de umidade relativa compreendidas na faixa de 20% a 95%, podendo operar em temperaturas que variam de 0°C a 60°C, com uma acurácia de 5% (AOSONG ELECTRONICS, 2018, tradução nossa).

Figura 11 - Sensor HR202L

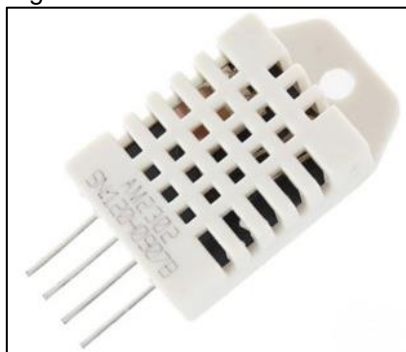


Fonte: Isweek (2018).

Entre os modelos existentes de sensores de umidade e temperatura, também podem ser encontrados alguns capazes de medir as duas variáveis simultaneamente, gerando maior facilidade de manuseio e menor espaço ocupacional. Como por exemplo, os modelos AM2302 e a DHT11.

O modelo AM2302 (figura 12) ou também conhecido por DHT22, possui capacidade de operar em faixas de temperatura de -40°C até +80°C e umidade variando entre 0% e 100%, com uma acuracidade de 0.5°C e 2%, respectivamente. Com relação a tensão em que o mesmo opera, está na faixa de 3.3V a 6V (AOSONG ELECTRONICS, 2018, tradução nossa).

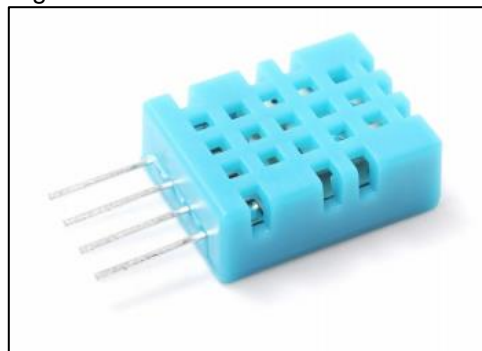
Figura 12 - Sensor AM2302



Fonte: Instituto NCB (2018).

Segundo as informações do *datasheet* fornecidas pela Mouser Electronics (2018, tradução nossa), o modelo DHT11 (figura 13), pode atuar em faixas de temperatura de 0°C a 50°C e umidade relativa de 20% a 90%, com uma precisão de 2°C para a temperatura e 5% para a umidade. Apesar da capacidade e acurácia de atuação deste sensor serem inferiores aos que foram citados anteriormente, a sua utilização em trabalhos acadêmicos ou em pequenos protótipos, possui uma boa viabilidade devido ao seu baixo custo de aquisição e fácil manuseio

Figura 13 - Sensor DHT11



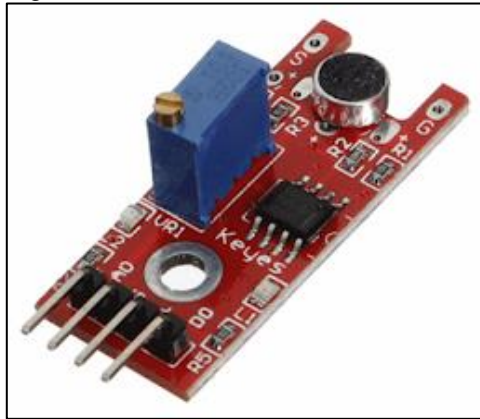
Fonte: Aosong (2019).

5.3 SENSORES DE CAPTAÇÃO DE RUÍDOS

Sensores de som são módulos eletrônicos criados com objetivo de captar e medir variações sonoras de um determinado ambiente. Um dos exemplos destes sensores encontrados no mercado, é o KY-038.

De acordo com as informações disponibilizadas pelo comércio de eletrônicos MasterWalker (2018), o modelo KY-038 (figura 14) tem como principal finalidade medir a intensidade sonora do ambiente em que está contido, ao qual varia a sua saída digital conforme a detecção de sons ao seu redor. Sua utilização é encontrada principalmente em projetos de automação residencial ou de segurança servindo por exemplo como alarme.

Figura 14 - Sensor KY-038



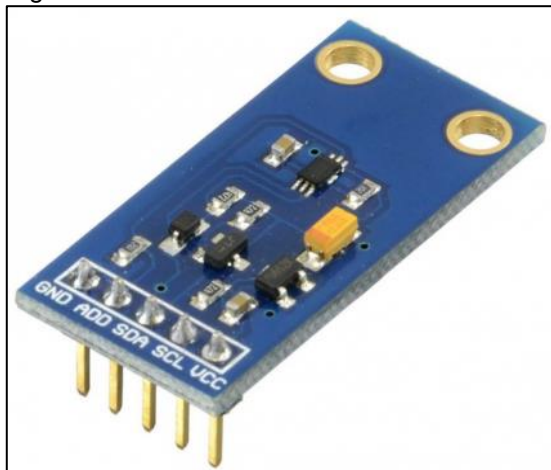
Fonte: Datasheetcafe (2018).

5.4 SENSORES DE LUMINOSIDADE

Os sensores de luz possuem como objetivo determinar a quantidade de luminosidade (medida em lux) que está incidindo sobre os mesmos, como exemplos destes sensores, pode ser citado os modelos BH1750FVI e o LDR.

O modelo BH1750FVI, de acordo com as informações do datasheet fornecidas pela Rohm Semiconductor (2011, tradução nossa), possui uma boa capacidade de detecção de luz, entre 1 até 65535 Lux, podendo operar em temperaturas que variam de -40°C a +85°C, como pode ser observado na figura 15.

Figura 15 - Sensor BH1750FVI



Fonte: Sunrom Electronics / Technologies (2018).

O modelo *Ligth Dependent Resistor* (LDR) representado na figura 16, é um componente que tem sua resistência alterada de acordo com a quantidade de luz em que é submetido, possuindo capacidade de trabalhar em temperaturas que podem

variar de -60°C até $+75^{\circ}\text{C}$, recebendo uma potência máxima de 100mW (SUNROM TECHNOLOGIES, 2008, tradução nossa).

Devido ao baixo custo de aquisição e facilidade operacional que este sensor possui, a sua utilização em pequenos protótipos que não requerem uma boa precisão, se torna uma opção de uma viabilidade considerável.

Figura 16 - Sensor LDR



Fonte: Wendling (2010).

5.5 CONEXÃO COM DISPOSITIVOS EXTERNOS

Em termos gerais, sistemas de comunicação é uma junção de mecanismos que possibilitam o processamento e o transporte de dados de uma determinada origem até seu destino (OLIVEIRA; LIMA, 2017).

Dentre as possíveis formas de comunicação entre microcontroladores, como por exemplo o Arduino, e dispositivos externos como *smartphones* ou até mesmo servidores, podem ser citadas as tecnologias *Bluetooth* e *Wi-Fi*. Ao qual nos próximos subcapítulos, serão detalhados o funcionamento e as características de cada uma delas.

5.5.1 Módulo Bluetooth

A tecnologia *Bluetooth* tem sua origem a partir da necessidade de uma conexão entre dispositivos móveis e outros acessórios pessoais, em ambientes desprovidos de conexão com a Internet e sem a necessidade de cabos, por meio da utilização de rádios de curto alcance, baixa potência e custo (PASSOS, 2011). Tendo sua utilização fortemente alavancada nos últimos anos devido a grande maioria dos dispositivos móveis usarem de seus recursos.

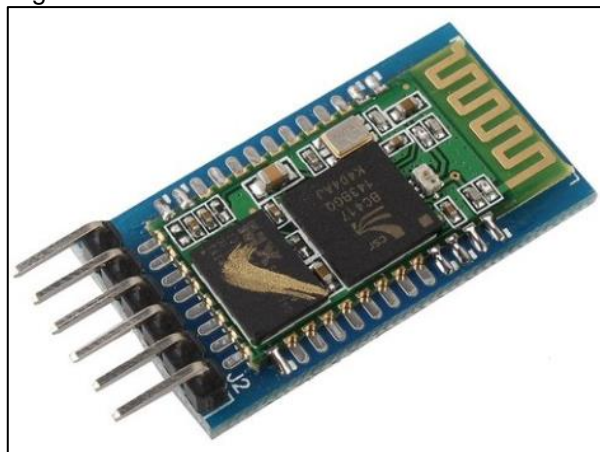
De acordo com Matos (2017), os dispositivos *Bluetooth* possuem a capacidade de encontrar outros próximos a ele, tendo a possibilidade de criar conexões independentes, sem precisar que o usuário faça algum tipo de intervenção. Conexões estas que podem ser conhecidas por mestre-escravo, na qual o dispositivo que efetua a mesma é chamado de mestre e os demais de escravos.

A fim de realizar as conexões entre dispositivos, o Bluetooth se utiliza da banda *Industrial, Scientific and Medical* (ISM) de 2,4 GHz, a uma frequência de comunicação de 54Mbps na versão 5.0, tendo capacidade de alcance variando de 1m e 50m (MATOS, 2017).

O mercado atual destes componentes, passíveis de conexão com microcontroladores como por exemplo Arduino, possui uma boa diversidade de modelos, dentre eles pode ser citado o HC-05 (figura 17).

De acordo com as informações do *datasheet* disponibilizado pela Future Eletronic Corporation (2018, tradução nossa), o módulo HC-05 pode operar em uma tensão de 5V, com alcance de 10 metros e pode suportar tanto o modo escravo, como o mestre, ou seja, tanto pode ser conectado, como conectar-se a outro dispositivo.

Figura 17 - Módulo HC-05



Fonte: Oddwires (2019).

5.5.2 Módulo Wi-Fi

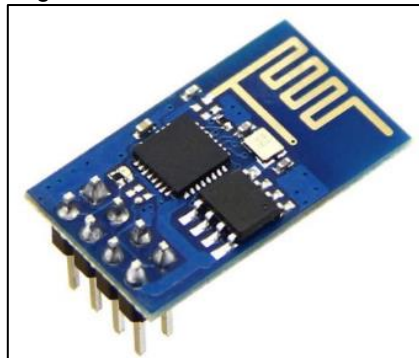
A tecnologia *Wi-Fi* teve origem através de uma marca licenciada pela *Wi-Fi Alliance*, que tem por objetivo descrever a tecnologia de redes sem fio embarcadas *Wireless Local Area Network* (WLAN) baseadas no padrão *Institute of Electrical and Eletronics Engineers* (IEEE) 802.11 (FERREIRA, 2013).

As redes de comunicação Wi-Fi, trabalham por meio de ondas de rádio, ao qual são geradas e transmitidas por um adaptador chamado roteador, que tem por finalidade receber os sinais, decodifica-los e emití-los para outros dispositivos que estejam dentro do seu raio de ação, nomeado de *hotspot* (LANDIM, 2012).

Com relação ao mercado atual de módulos Wi-Fi, compatíveis de conexão com o Arduino, pode ser citado o modelo ESP8266 (figura 18).

De acordo com as informações providas pelo *datasheet* fornecido pela Espressif Systems (2013, tradução nossa), este componente suporta as redes 802.11 b/g/n, podendo operar como um ponto de acesso ou em forma de uma estação, enviando e recebendo informação a uma distância que pode chegar em torno de 90m de acordo com cada ambiente.

Figura 18 - Módulo Wi-Fi ESP8266



Fonte: Fingerpointengg (2018).

Dentre as tecnologias de comunicação citadas anteriormente, que são o *Bluetooth* e o *Wi-Fi*, ambas possuem suas características consideravelmente distintas. Cada qual com suas vantagens e desvantagens, dependendo principalmente do contexto em que cada uma delas são empregadas, onde para o *Bluetooth* é possível a realização de conexões locais e já o *Wi-Fi* disponibiliza a possibilidade de acesso à *Internet* (FERREIRA, 2013).

6 TRABALHOS CORRELATOS

Trabalhos correlatos são estudos já realizados em uma outra oportunidade e que possuem alguma relação com o referente trabalho. A seguir nos próximos subcapítulos serão apresentados alguns deles.

6.1 DESIGN AND IMPLEMENTATION OF PIG FARM MONITORING SYSTEM FOR UBIQUITOUS AGRICULTURE

Estudo realizado por Ho-chul Lee e Hyun Yeo, pelo curso de Engenharia de Informação e Comunicação na Universidade Nacional de Sunchon, cidade de Sunchon localizada na Coréia do Norte, no ano de 2010. Tendo como objetivo a implementação de um sistema de monitoramento de suínos, visando a maximização da produção através da instalação de sensores de temperatura, umidade e câmera de vídeo, aos quais foram transmitindo em tempo real para os suinocultores.

De acordo com Lee e Yeo (2010, tradução nossa), o sistema funcionava de tal forma, que quando o ambiente em que os sensores estavam contidos, perdesse o nível de conforto térmico para a criação de suínos, equipamentos de controle eram acionados, como umidificador, ar condicionado e janelas automáticas, com isso mantendo valores ambientalmente corretos e uniformes.

Com a implementação realizada, duas formas de controle do ambiente ideal são disponibilizadas aos produtores, a primeira delas é deixar que o sistema atue automaticamente com as configurações pré-estabelecidas. Já o segundo se refere a uma intervenção manual, onde nela o sistema já adotará de forma automática os novos padrões informados de conforto térmico, com isso na próxima execução do mesmo, já estarão sendo executado com base nestas informações (LEE; YEO, 2010, tradução nossa).

Segundo Lee e Yeo (2010, tradução nossa), os resultados da pesquisa foram positivos, ao qual pode-se concluir que o local de aplicação do sistema em questão, foi corretamente monitorado e controlado para possuir condições ideais a criação de suínos, como também recebendo um *feedback* positivo por partes dos produtores referente ao sistema de câmeras em monitoramento *online*.

6.2 AUTOMAÇÃO RESIDENCIAL DE BAIXO CUSTO UTILIZANDO ARDUINO

Monografia realizada pelo acadêmico Tiago Melillo de Mattos, do curso de Engenharia de Controle e Automação da Universidade Federal de Ouro Preto, no ano de 2017. Possui como principal objetivo a realização de um estudo para a implementação de automações residenciais, utilizando-se do microcontrolador Arduino, visando o baixo custo.

Com o expressivo avanço tecnológico ocorrido nos últimos anos, proporcionou que componentes eletrônicos avançados ficassem mais acessíveis as pessoas, tanto com relação a custo, como em facilidade de programa-los. Com isso, permitindo que a popularização na realização de automações residenciais, mantenha-se em constante crescimento (MATTOS, 2017).

No trabalho em questão, Mattos (2017) realizou uma pesquisa referente a variados tipos de sensores, passíveis de aplicação em cada situação em que se pretendia automatizar dentro de uma residência. Como também, enumerou as possíveis formas de comunicação sem fio, possíveis de aplicação em um futuro protótipo a ser desenvolvido, como *Wi-Fi* e *Bluetooth*.

Dentre as possibilidades de automatização residencial pesquisadas por Mattos, estão:

- a) iluminação: aplicação de sensores instalados na parte externa da residência, acionando lâmpadas ao anoitecer ou até mesmo abrindo cortinas motorizadas ao detectar a luz do dia;
- b) temperatura: implementação de sensores, para o controle de ar-condicionado de acordo com a temperatura ambiente, a fim de manter o bem-estar das pessoas que habitam a residência;
- c) presença: aplicação de sensores para o acionamento de lâmpadas ao detectar-se a presença de indivíduos;
- d) controle de incêndio: aciona irrigadores de parede, ao se detectar a presença de gases inflamáveis e fumaça;

De acordo com Mattos (2017), os objetivos do trabalho, referente a introdução dos conceitos necessários para o desenvolvimento de um projeto de automação residencial de baixo custo, foram alcançados. Visto que, foram enumeradas as principais tecnologias possíveis de aplicação em um futuro protótipo.

6.3 ESTUDO DE FRAMEWORK MULTIPLATAFORMA APLICADO AO DESENVOLVIMENTO DE UM PROTÓTIPO DE APLICATIVO MOBILE HÍBRIDO PARA O CONTROLE FINANCEIRO PESSOAL

Em 2015, o acadêmico Muriel Rampinelli Mazuchetti, do curso de Ciência da Computação da Universidade do Extremo Sul Catarinense - UNESC, realizou um estudo referente a viabilidade da aplicação de *frameworks* no desenvolvimento de aplicações mobiles, em específico o Ionic. Ao qual aplicou na prática a criação de uma aplicação para o controle de finanças pessoais.

Com o crescimento expressivo na utilização de dispositivos móveis, como celulares, tablets e *smartphones*, onde cada vez mais estão se tornando uma peça chave na vida das pessoas, é notável a vasta quantidade de vantagens e benefícios de fácil acesso disponibilizados aos usuários. Com isso, fazendo com que os desenvolvedores cada vez mais, busquem aperfeiçoar seus conhecimentos referentes a este avanço e as novas tendências do mercado (MAZUCHETTI, 2015).

Diante desta situação, surgiram no mercado novas ferramentas que auxiliam o desenvolvimento de aplicações, reduzindo custos de desenvolvimento, número de manutenções e a necessidade de reescrita do código fonte para cada plataforma específica. Dentre as opções disponíveis e com base nas pesquisas realizadas optou-se pela utilização do *framework* Ionic (MAZUCHETTI, 2015).

Segundo Mazuchetti (2015), o *framework* Ionic pode ser entendido como uma junção de tecnologias e utilitários a fim de otimizar o desenvolvimento de aplicações móveis híbridas, sendo estas o HTML, CSS e *JavaScript*. Para que o mesmo possa ser executado, é necessário também de outras ferramentas que o compõem, como o AngularJS sendo o *framework* de aplicação web JavaScript, e o Cordova, ficando como responsável pela construção e interpretação do aplicativo para as plataformas móveis.

De acordo com Mazuchetti (2015), os resultados obtidos com a pesquisa e aplicação do Ionic, para o desenvolvimento de uma aplicação para controle de finanças pessoais são satisfatórios. Uma vez que, as funcionalidades deste, apresentaram um diferencial com relação aos demais *frameworks*, facilitando o aprendizado e o desenvolvimento de uma aplicação de forma ágil e dinâmica.

6.4 DESENVOLVIMENTO DE SISTEMA AUTOMATIZADO DE MONITORAMENTO DE AMBIENTES DE PRODUÇÃO ANIMAL, UTILIZANDO UMA REDE DE SENSORES SEM FIO

Dissertação produzida pelo mestrando Marcelo Eduardo de Oliveira, na Faculdade de Zootecnia e Engenharia de Alimentos da Universidade de São Paulo, no ano de 2015. Tem como principal objetivo, o desenvolvimento e testes de um sistema automatizado para a realização de um controle ambiental, utilizando-se de sensores sem fio, proporcionando uma melhor segurança no controle destes ambientes.

O cenário escolhido para a realização deste monitoramento, deu-se em um ambiente de produção animal, em específico o de aves. Aos quais foram monitoradas as variáveis de temperatura, umidade e luminosidade, tendo estas, uma influência direta na produtividade apresentada pelos animais.

No projeto, utilizou-se do microcontrolador Arduino como centralizador das informações, ao qual são conectados nele os sensores específicos para a leitura de cada variável. Onde para a captura da luminosidade, utilizou-se do sensor LDR e para a temperatura e umidade, foi utilizado o DHT22.

Com a coleta dessas informações, torna-se possível a inferência do sistema sobre o ambiente que está sendo monitorado, através de equipamentos como ventiladores, aquecedores elétricos, sistema de nebulização e cortinas automatizadas, aos quais são controladores pelo Arduino por meio de um Relé.

Além da utilização do Arduino, foi utilizado o protocolo ZigBee para o gerenciamento do fluxo de dados do sistema. Visto que, este proporciona a possibilidade de que uma rede sem fio seja capaz de fazer com que seus nós, retransmitam as informações dos demais, até que estas informações cheguem ao seu destino (OLIVEIRA, 2015).

Como resultados da pesquisa de acordo com Oliveira (2015), o sistema desenvolvido trouxe ótimos resultados, visto que simplificou as tarefas de instalação e manutenção, como também aumentando o nível de automação no ambiente de produção animal.

Concluiu-se também, que a utilização da tecnologia ZigBee teve bons resultados na aplicação em um ambiente de produção animal, visto que os dados obtidos tiveram ótima precisão e confiabilidade (OLIVEIRA, 2015).

6.5 SISTEMA AUTOMATIZADO DE BAIXO CUSTO PARA PRODUTORES RURAIS: CONTROLE E MONITORAMENTO DO AMBIENTE TÉRMICO NA SUINOCULTURA

Pesquisa realizada por Pedro Hurtado de Mendoza Borges, Zaíra Moraes dos Santos Hurtado Mendoza, Pedro Hurtado de Mendoza Moraes e Ronei Lopes dos Santos no ano de 2018 e publicada na Revista Eletrônica Competências Digitais para Agricultura Familiar.

Possui como principal objetivo, o desenvolvimento de um sistema de baixo custo, para o monitoramento e controle das variáveis de temperatura e umidade de um ambiente de produção de suínos.

Para a realização deste monitoramento, utilizou-se do microcontrolador Arduino como gerenciador e centralizador das informações, onde neste foi acoplado o sensor específico para a leitura dos dados de temperatura e umidade, ao qual no projeto utilizou-se do modelo DHT22.

Após a coleta dos dados realizada, implementou-se a utilização de equipamentos como ventiladores e nebulizadores a fim de manter o nível térmico adequado. Aos quais estes podendo ser controlados pelo próprio Arduino, utilizando-se de um Relé.

A fim de proporcionar o gerenciamento e visualização dos dados coletados, implementou-se também a utilização de um módulo de cartão SD, ao qual são armazenadas todas as verificações realizadas pelo microcontrolador em um arquivo no formato de texto.

De acordo com Borges et al. (2018), os resultados obtidos com a pesquisa, comprovam que é possível realizar a automação de baixo custo de processos relacionados a suinocultura, em pequenas propriedades rurais. Ao qual comprovou-se que a utilização de microcontroladores integrado a sensores pode colaborar substancialmente para o aumento do conforto térmico do animal.

Chegou-se à conclusão também, de que o consumo de energia e água nas propriedades em que o sistema for implantado, pode diminuir consideravelmente, visto que os equipamentos são acionados, apenas nos períodos necessários e tomando como base critérios técnicos pré-definidos (BORGES et al., 2018).

7 AUTOMATIZAÇÃO E MONITORAMENTO DE PROCESSOS RELACIONADOS AO BEM-ESTAR ANIMAL DENTRO DA CADEIA PRODUTIVA DA SUINOCULTURA POR MEIO DE MICROCONTROLADORES INTEGRADOS A UMA APLICAÇÃO MOBILE

Baseando-se em todo referencial teórico coletado e no conhecimento adquirido a partir dele, o trabalho em questão possui como finalidade a pesquisa e o desenvolvimento de um protótipo para o monitoramento de variáveis relacionadas ao bem-estar animal no ramo da suinocultura.

A fim de realizar a coleta dos dados, implementou-se a utilização do microcontrolador Arduino, integrado a sensores específicos para cada situação, onde o mesmo também tem como responsabilidade, o envio dos dados coletados para um servidor na nuvem. Por fim, para o gerenciamento dessas informações, foi realizado o desenvolvimento de uma aplicação mobile, utilizando-se do framework Ionic.

7.1 METODOLOGIA

Para a realização deste projeto de pesquisa, a primeira etapa exercida foi referente ao levantamento bibliográfico, a qual foi efetuada por meio de artigos, livros, publicações e pesquisas em geral na Internet.

Com relação as bases de dados pesquisadas na Internet, podem ser citadas: Google Acadêmico; ABCS e EMBRAPA relacionadas a materiais voltados para a produção de suínos; IEEE Xplore e em repositórios de trabalhos acadêmicos de diversas Universidades.

Após isso, iniciou-se propriamente dito a escrita do referencial teórico, apresentando os principais assuntos que norteiam os procedimentos necessários para o desenvolvimento do protótipo proposto.

Nesta etapa foram abordados temas como: produção de suínos e a importância do manejo do bem-estar animal, dando maior ênfase as variáveis de monitoração selecionadas; microcontroladores e sensores para a realização da coleta dos dados; métodos de desenvolvimento mobile com maior destaque na utilização do framework Ionic e meios para a comunicação de dados entre essas tecnologias.

Para a escolha de quais variáveis de monitoração seriam compreendidas no projeto, baseou-se em pesquisas realizadas em materiais de produção de suínos

fornecidos pelo EMBRAPA e ABCS, ao qual avaliou-se quais as possibilidades que estavam em pauta e que contêm importância fundamental na cadeia produtiva dos suínos. Como também, a disponibilidade de sensores para o monitoramento e que fossem passíveis de integração com o microcontrolador selecionado.

As demais etapas estão baseadas no desenvolvimento do protótipo, iniciando-se primeiramente pelo levantamento dos recursos necessários, após isso estipulou-se o fluxo de dados e a modelagem do banco de dados, seguidos pela configuração do servidor responsável pelo armazenamento dos dados, montagem da simulação com a estruturação do microcontrolador e demais dispositivos, desenvolvimento da aplicação e por fim os testes de comunicação dos dados entre todas estas etapas e funcionalidades gerais do projeto.

Com a realização de todos estes estágios anteriormente descritos, torna-se possível produzir uma pesquisa de natureza aplicada de base tecnológica, tendo como abordagem o modelo quantitativo.

7.1.1 Ferramentas e recursos utilizados

Com relação às ferramentas utilizadas no projeto, iniciou-se a busca tendo como necessidade a modelagem do banco de dados, onde utilizou-se do MySQL Workbench¹ na versão 8.0.15. Sendo esta, uma ferramenta visual unificada para arquitetos de banco de dados e desenvolvedores. Fornece modelagem de dados, desenvolvimento de SQL e ferramentas de administração abrangentes. Disponível gratuitamente para os SOs Windows, Linux e Mac OSX (MYSQL WORKBENCH, 2019, tradução nossa).

Para a programação do microcontrolador Arduino, foi utilizada o Arduino IDE², versão 1.8.8, fornecida pelo próprio site oficial do equipamento, disponível para download gratuitamente, estando acessível para os SOs Windows, Linux e Mac.

Para o desenvolvimento da aplicação e demais códigos fontes necessários, foi utilizado o Visual Studio Code³, versão 1.35.0. Ferramenta esta, mantida pela Microsoft e disponível para Windows, Linux e Mac. Ao qual fornece suporte para as

¹ Site: <https://www.mysql.com/products/workbench>

² Site: <https://www.arduino.cc>

³ Site: <https://code.visualstudio.com>

principais linguagens de programação disponíveis no mercado atual, podendo ser adquirida gratuitamente.

Ainda com relação ao desenvolvimento da aplicação, foi necessário a configuração do ambiente para a execução do Ionic, tendo a necessidade de instalação das ferramentas NodeJS⁴ e npm⁵, aos quais foram utilizadas as versões 10.14.2 e 6.4.1 respectivamente.

O NodeJS pode ser entendida como um ambiente de execução, tendo sua construção baseada no motor JavaScript do Google Chrome, ao qual permite que o JavaScript seja executado no lado do servidor (IONIC, 2019, tradução nossa). Já o npm tem como objetivo ser um gerenciador de pacotes, onde através de linhas de comando, serve como um utilitário para a interação com repositórios *onlines*, provendo gerenciamento de versão e dependências e auxílio na instalação de pacotes (IONIC, 2019, tradução nossa).

Através do npm, torna-se possível via linha de comando a instalação do Ionic, conforme orientações contidas no apêndice A, obtendo-se a versão 4.10.2 sendo umas das mais atuais, visto que a versão quatro teve lançamento em janeiro de 2019. Com isso, o usuário passa a ter acesso a todos os recursos do Ionic CLI, tendo este a capacidade de oferecer uma ampla gama de ferramentas de desenvolvimento, compilação, depuração de códigos e opções de ajuda ao longo de sua utilização (IONIC, 2019, tradução nossa).

Para concluir as configurações do Ionic, utilizou-se do Android SDK Manager⁶, sendo este responsável por disponibilizar as versões do sistema operacional Android para download, sendo utilizado no projeto a API 25. Nestas contendo ferramentas para o desenvolvimento, bibliotecas e emuladores necessários para a criação de uma aplicação com intuito de rodar no sistema operacional Android.

Com relação a necessidade de armazenamento dos dados do protótipo, utilizou-se um servidor na nuvem. Ao qual para a decisão da melhor opção que se adequaria ao projeto, os principais quesitos levados em consideração, foram o custo de hospedagem e os recursos disponibilizados por cada uma das opções.

⁴ Site: <https://nodejs.org>

⁵ Site: <https://www.npmjs.com/get-npm>

⁶ Site: <https://developer.android.com/studio>

Como resultado dessa pesquisa, encontrou-se o *000webhost*⁷, tendo este a possibilidade de uso de forma gratuita e disponibilidade de vários recursos, dentre eles um gerenciador de arquivos, PHP 7.1 e banco de dados relacional MySQL previamente configurados, juntamente com o painel avançado PhpMyAdmin para o gerenciamento deste último.

De acordo com Souza e Viana (2016), o MySQL é uma ferramenta de código aberto, possuindo uma boa popularidade devido ao baixo custo e confiabilidade para sua utilização. Como também, tendo uma boa escalabilidade e desempenho, atendendo a necessidades de projetos com maior fluxo de dados. Para o servidor em questão o mesmo contém um limite máximo de 150 tabelas e 1 GB de armazenamento para a versão gratuita.

O MySQL também pode ser entendido como um Sistema de Gerenciamento de Banco de Dados (SGBD), que se utiliza de uma linguagem de consulta estruturada, do inglês Structured Query Language (SQL) como interface para inserir, acessar e gerenciar todo o conteúdo nele armazenado.

Com respeito a montagem da simulação proposta para a realização da captura dos dados, foram utilizados uma série de equipamentos, aos quais estão listados na tabela 5.

Tabela 5 - Equipamentos utilizados para a montagem da simulação

Descrição	Qtd	Função	Valor: US\$ Ref.: 03/2019
Placa MEGA 2560 R3 + Cabo USB para Arduino	1	Centralizar / controlar todo o fluxo de dados e demais equipamentos envolvidos	24,15
Sensor de Umidade e Temperatura DHT11	1	Capturar os valores de temperatura e umidade	2,78
Sensor de Luminosidade LDR 5mm	1	Capturar os valores de luminosidade	0,48
Sensor de Som KY-038 Microfone	1	Capturar os valores de sonoridade	2,26
Módulo Relé 5V 2 Canais	1	Acionamento dos equipamentos	3,28
Jumpers	40	Fios de conexão entre equipamentos	2,26
Protoboard 830 Pontos	1	Acoplar os sensores e disponibilizar carga necessária aos mesmos	4,30

⁷ <https://www.000webhost.com>

Descrição	Qtd	Função	Valor: US\$ Ref.: 03/2019
Módulo WiFi ESP8266 ESP-01	1	Envio dos dados coletados ao servidor da nuvem	6,34
Adaptador para Módulo WiFi ESP8266 ESP-01	1	Converter a tensão de 5V do Arduino para 3.3V do módulo ESP8266	5,06
Fonte DC Chaveada 9V 1A Plug P4	1	Converte a tensão de entrada de 220V para 9V suportada pelo Arduino	3,79
Lâmpadas Incandescente 40w	1	Simular o aquecimento do ambiente	2,54
Cooler 5V	2	Simular o resfriamento do ambiente	4,52
Total			61,76

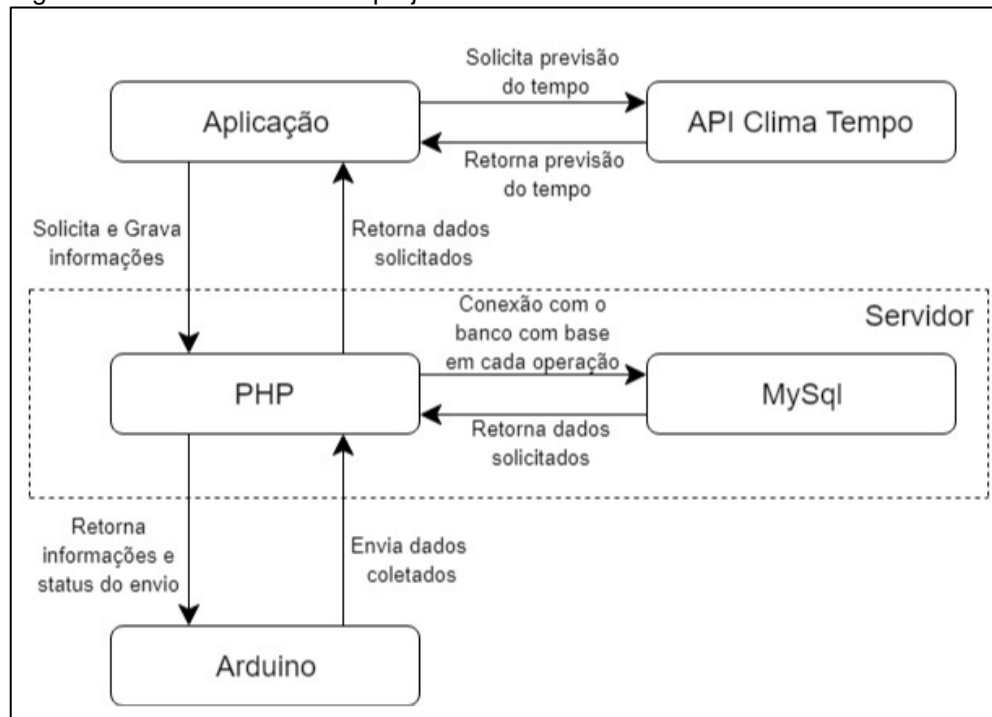
Fonte: Do autor.

De acordo com os itens apresentados na tabela 5, torna-se possível realizar a montagem do circuito, para verificação das variáveis de ambientes relacionadas ao bem-estar animal propostas no projeto.

7.1.7 Fluxo dos dados e modelo de comunicação

Para o início do desenvolvimento, primeiramente estipulou-se um diagrama de fluxo de dados (figura 19), tendo como intuito o mapeamento de todos os processos que o projeto é contido e todas as interconexões necessárias entre os mesmos. Com isso, possibilitando uma visão global, facilitando a percepção de todas as etapas a serem realizadas.

Figura 19 - Fluxo de dados do projeto



Fonte: Do autor.

O fluxo inicial estipulado, acontece no Arduino, onde este deve reunir as informações necessárias e realizar o envio para o banco de dados no servidor na nuvem. Sendo que, esta conexão é intermediada por um arquivo PHP localizado no próprio servidor, tendo este o objetivo de centralizar todas as operações que envolvem de alguma forma o uso da base de dados.

O PHP é uma linguagem de *script open source* de uso geral, altamente popularizada, e principalmente adequada para o desenvolvimento web. Possui suporte para a maioria dos sistemas operacionais presentes no mercado atual, e tem como uma de suas características a possibilidade de programação estruturada ou orientada a objeto, ou até mesmo uma junção de ambos os formatos (PHP, 2019).

Por fim, a aplicação é a última parte do fluxo, ao qual nesta será possível realizar o gerenciamento das informações coletadas. Onde para a conexão com a base de dados, segue o mesmo formato citado anteriormente via arquivo PHP. Esta etapa também possui, a comunicação com uma API externa para a obtenção de previsão do tempo, sendo uma funcionalidade extra do projeto.

Com relação ao formato de comunicação entre todas estas etapas, se deu por meio de requisições HTTP, utilizando-se dos métodos GET e SET dependendo de cada ocasião.

O Protocolo de Transferência de Hipertexto, do inglês *HyperText Transfer Protocol* (HTTP), se baseia no formato requisição-resposta, trabalhando no modelo computacional cliente-servidor. Sendo composto por métodos ou também conhecidos por verbos, tendo estes a função de identificar qual a ação a ser executada, fazendo-se necessário especificar qual será utilizado para a realização de uma requisição (DEVMEDIA, 2012). Dentre os mais utilizados, estão o GET e o POST.

O GET tem em sua utilização, o objetivo principal de realizar requisições de dados a um determinado serviço diretamente via URL, como por exemplo, o processamento de um browser a um link informado por um usuário. Já o método POST, possui uma finalidade contrária, ao qual deve ser usado para o envio ou submissão de dados ao servidor. Tendo como diferença também, o fato de que o POST não transmite os dados diretamente via URL, com isso não ficando visível ao cliente, no qual aumenta o nível de segurança durante a conexão (DEVMEDIA, 2012).

Já com relação ao formato dos dados envolvidos em cada transmissão, foi definido a padronização do uso do *JavaScript Object Notation* (JSON), para todas as operações realizadas, tendo esta metodologia o objetivo principal de ser uma forma ágil na comunicação entre sistemas.

De acordo com Silva (2018), o formato de dados JSON consiste na formação de pares de atributos-valor e matrizes, sendo uma ótima forma para realizar a troca de informações entre sistemas, ao qual deriva do JavaScript.

Outras características quanto ao formato JSON, estão relacionadas a maior velocidade na execução e transporte dos dados, analisador e leitura mais simples, suporte a informações no formato de objetos e arquivos com tamanhas reduzidos (DEVMEDIA, 2011).

A única etapa que não segue este padrão de comunicação mencionado, refere-se as conexões internas dentro do servidor realizadas pelo PHP e o banco de dados MySQL, representados pela área pontilhada na figura 19. Visto que, esta comunicação é realizada diretamente por funções do próprio PHP, onde através destas, são enviados ao banco, os comandos SQL necessários e este processa e retorna as informações de acordo com cada situação.

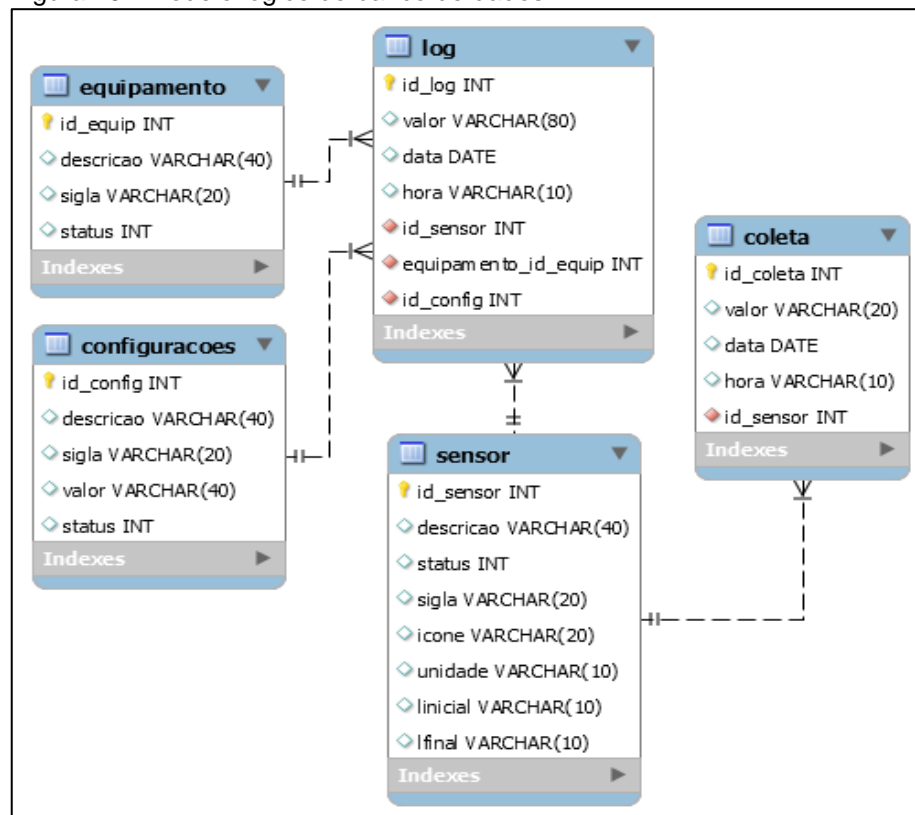
Com o fluxo de dados definido e tendo uma ideia da abrangência de todo o projeto, torna-se possível realizar a modelagem do banco de dados.

7.1.1 Modelagem dos dados

A modelagem do banco de dados, facilita a organização e levantamento de todos os recursos necessários previamente ao desenvolvimento, visto que, ela permite junto ao fluxo de dados, uma percepção geral de todo o conjunto de informações do projeto.

A figura 20 ilustra o modelo lógico do banco de dados do projeto.

Figura 20 - Modelo lógico do banco de dados



Fonte: Do autor.

Conforme ilustrado na figura 20, a modelagem é composta por cinco tabelas, que são:

- equipamento: responsável por armazenar os equipamentos que podem ser acionados via aplicação para o controle das variáveis de bem-estar animal, como os coolers e a lâmpada, todos eles com seu status atual de ligado ou desligado;

- b) configuracoes: tabela genérica para armazenar todas as configurações do projeto, como por exemplo, o intervalo de tempo de verificação do microcontrolador e o status atual da opção de automatização;
- c) sensor: nesta é armazenado o cadastro de todos os sensores utilizados no projeto;
- d) coleta: responsável por armazenar um histórico de todas as verificações realizadas pelo microcontrolador individualmente por sensor, com sua respectiva data e horário;
- e) log: armazena todos os logs de alterações realizados pelo usuário na aplicação, seja pelo acionamento de equipamentos ou em modificações feitas nas configurações do aplicativo.

Com a modelagem do banco de dados concluída, realizou-se um levantamento de todas as ferramentas e recursos necessários, que seriam de alguma forma, utilizados para o desenvolvimento e montagem do protótipo.

7.1.3 Configuração do servidor

Conforme já comentado anteriormente no tópico de ferramentas e recursos utilizados, o servidor escolhido para o armazenamento dos dados na nuvem, foi o 000webhost, devido ao custo e benefícios que o mesmo proporciona.

A primeira etapa relacionada ao uso deste ambiente, está atrelada a criação das tabelas no banco de dados MySQL, ao qual realizou-se por meio do painel de gerenciamento PhpMyAdmin, onde este disponibiliza opções para a edição e execução de comandos SQL.

O SQL é uma linguagem padronizada universalmente, que possui o objetivo de acessar e manipular bancos de dados por meio de SGBDs. A figura 21 ilustra um exemplo de comando SQL para a criação da tabela de equipamento utilizada projeto.

Figura 21 - Criação das tabelas via comando SQL

```
CREATE TABLE equipamento (  
id_equip INT(6) AUTO_INCREMENT PRIMARY KEY,  
descricao VARCHAR(40),  
sigla VARCHAR(20),  
status int(1))
```

Fonte: Do autor.

Posteriormente a criação das tabelas, o uso do painel PhpMyAdmin torna possível o gerenciamento das informações já incluídas na base de dados, conforme ilustrado na figura 22, onde pode-se visualizar, alterar e excluir os registros individualmente de cada tabela.

Figura 22 - Painel de gerenciamento

✓ A mostrar registos de 0 - 1 (2 total, A consulta demorou 0.0048 segundos.)

`SELECT * FROM `equipamento``

☐ Perfil [Editar em linha] [Edita] [Explicar SQL] [Criar código PHP] [Atualizar]

☐ Mostrar tudo | Número de registos: 25 | Filtrar registos: O

+ Opções

	id_equip	descricao	sigla	status
<input type="checkbox"/> Editar Copiar Apagar	1	VENTILADOR	VEN	0
<input type="checkbox"/> Editar Copiar Apagar	2	LÂMPADA	LUZ	0

☐ Marcar todos Com os seleccionados: Editar Copiar Apagar Exportar

Fonte: Do Autor.

A fim de disponibilizar um meio para realização da comunicação externa com a base de dados criada, como por exemplo através da aplicação e do microcontrolador utilizados no projeto, optou-se pelo uso do PHP.

A estrutura básica para a programação de *scripts* em PHP, consiste basicamente na abertura e fechamento da *tag* PHP, dada por `<?php` e `?>` (figura 23), e dentro desta, deve conter toda a programação necessária, podendo ser utilizado todos os recursos fornecidos pela linguagem.

Figura 23 - Estrutura básica do PHP

```
<?php
|   echo "Hello World";
?>
```

Fonte: Do Autor.

Já para a realização da comunicação com o banco de dados, o PHP fornece algumas alternativas, dentre elas a utilização da função `mysqli_connect`, onde

são passados os seguintes parâmetros: localização do banco, sendo o *localhost*, devido ao PHP estar situado no mesmo ambiente que o banco; usuário; senha e o nome da base de dados. Sendo que, esses três últimos são definidos no momento em que o banco é criado no servidor.

Posteriormente, é utilizado a função *mysqli_query* para realizar as operações desejadas no banco, como consultas, alterações e exclusões de registros. Onde através desta, são passados os dados de acesso ao banco, definidos anteriormente pela função *mysqli_connect* e o comando SQL que se deseja aplicar.

A fim de obter os resultados destas operações, pode-se utilizar da *mysqli_fetch_assoc*, onde nesta é possível capturar todos os dados retornados em uma consulta por exemplo. Por fim, usa-se a *mysqli_close* para o encerramento da conexão com o banco de dados.

A figura 24 ilustra a utilização das funções de acesso a base de dados.

Figura 24 - Funções de acesso a base de dados

```
$connect = mysqli_connect("localhost", $user, $senha, $banco);
$query   = 'select equipamento.sigla, equipamento.status from equipamento';
$result  = mysqli_query($connect,$query) or
          die('{"verro":"true","vretorno":"ERRO NO COMANDO SELECT DOS EQUIPAMENTOS"}');

while($retorno=mysqli_fetch_assoc($result)){

    $equip = $equip.'','.$retorno['sigla'].':'.$retorno['status'].'';
}

mysqli_close($connect);
```

Fonte: Do autor.

Para a conexão com o PHP, o mesmo foi desenvolvido com a ideia de que os processos de origem, devem realizar a comunicação utilizando-se das diretivas do protocolo HTTP. Ao qual, o desenvolvimento deste, foi estruturado de tal forma que pode operar tanto com a utilização do método GET como POST, ao qual tem como premissa básica, o recebimento dos dados no formato JSON, contendo o tipo de operação, que possibilitará o PHP identificar qual a solicitação desejada, e com isso executando rotinas específicas e as chaves de acesso ao banco de dados.

A figura 25 ilustra a formatação padrão do JSON de dados a ser enviado ao PHP desenvolvido.

Figura 25 - JSON utilizado no PHP

```
{
  "tipo": "1",
  "conexao": [
    {
      "user": "id_usuario",
      "senha": "XXX",
      "banco": "id_banco"
    }
  ]
}
```

Fonte: Do autor.

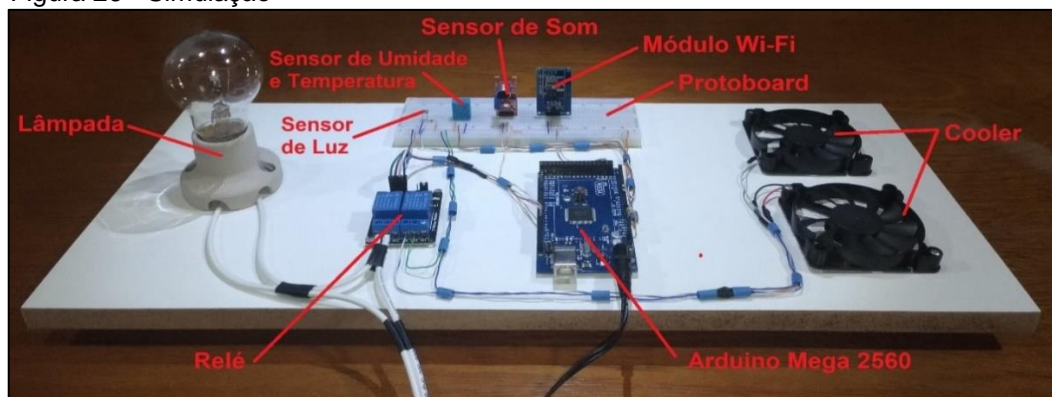
Após a conclusão do desenvolvimento de todas as operações necessárias no arquivo PHP, para finalizar a configuração do servidor, basta realizar o *upload* do arquivo, utilizando-se da ferramenta de gerenciador de arquivos disponibilizado no *000webhost*.

Com o término da configuração do servidor e da estrutura necessária para o armazenamento das informações do banco de dados, iniciou-se a confecção da simulação, utilizando-se dos equipamentos e recursos mencionados anteriormente para a coleta e envio dos dados para a nuvem.

7.1.4 Montagem da simulação

O primeiro passo efetuado nesta etapa, foi propriamente a montagem física da maquete, ao qual foram acoplados em um suporte todos os equipamentos necessários, com isso, realizando as interconexões necessárias entre os mesmos, conforme o esquema ilustrado na figura 26.

Figura 26 - Simulação



Fonte: Do autor.

Dentre os equipamentos utilizados, iniciou-se a montagem pelo protoboard, tendo este a responsabilidade de ser o suporte para a conexão entre os sensores e o microcontrolador. Como características principais, possui 830 furos, podendo operar a uma tensão máxima de 500V AC por minuto e a uma faixa de temperatura variando de -20°C a 80°C.

A próxima etapa se deu na configuração dos sensores utilizados para a coleta das variáveis de bem-estar animal, tendo como primeiro deles o DHT11, que possui objetivo de verificar a temperatura e umidade.

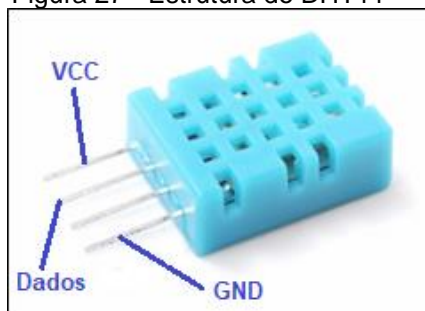
De acordo com os dados fornecidos pelo *datasheet* disponibilizados pela Aosong (2019, tradução nossa), este sensor possui a capacidade de coletar dados nas faixas de temperatura de 0°C a 50°C e umidade relativa de 20% a 90%, tendo uma precisão de $\pm 2.0^\circ\text{C}$ e $\pm 5\%$ respectivamente.

Outras características do DHT11, estão relacionadas ao tempo de resposta que o mesmo possui, que é de aproximadamente 2 segundos, e também com relação a alimentação necessária para o seu funcionamento, que pode variar de 3 a 5.0V e a uma corrente máxima de 2.5mA (AOSONG, 2019, tradução nossa).

Em algumas ocasiões, este sensor também pode ser conhecido pelo termo *Negative Temperature Coefficient* (NTC), sendo estes, dispositivos com resistência inversamente proporcional à temperatura, visto que, são dispositivos onde a resistência elétrica e a temperatura são conhecidas, onde quanto maior a temperatura menor será a resistência, possuindo assim, uma empregabilidade ideal em ambientes que requerem uma alta sensibilidade com mudanças climáticas (BILMAIA, 2018).

O DHT11, é formado por quatro terminais (figura 27), ao qual destes são utilizados somente três, que são o GND (terra), VCC (alimentação) e o de dados que é o responsável por transmitir as informações coletadas e deve ser conectado a uma porta analógica do microcontrolador.

Figura 27 - Estrutura do DHT11

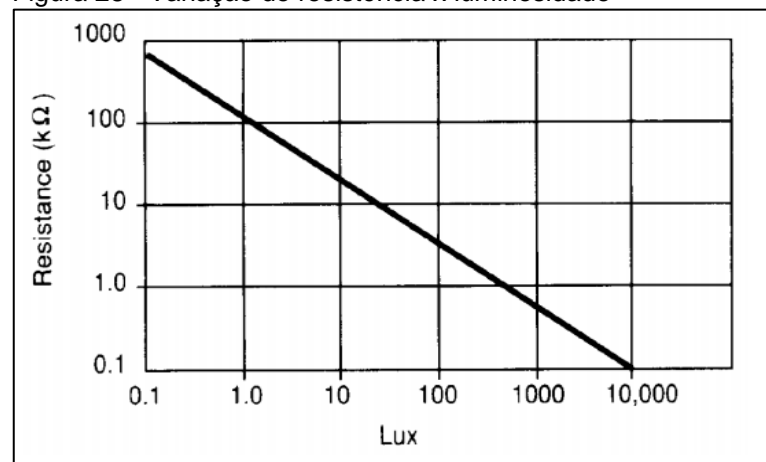


Fonte: Adaptado de Aosong (2019).

Com relação a captura dos dados de luminosidade, utilizou-se do sensor LDR. Com base nas informações disponibilizadas no datasheet fornecido pela Sunrom Technologies (2008, tradução nossa), este sensor possui a capacidade de alterar sua resistência com base na intensidade da luz que incide sobre ele, ao qual, quanto maior o nível de luminosidade, menor será a resistência.

O gráfico ilustrado na figura 28, representa a variação do valor da resistência de acordo com a intensidade da luz capturada pelo sensor LDR, ao qual pode-se observar que quanto maior é a luminosidade representada pelo eixo x do gráfico, menor é a resistência no eixo y.

Figura 28 - Variação de resistência x luminosidade



Fonte: Sunrom Technologies (2008).

Outras características que podem ser citadas referentes a este sensor, são: baixo custo de aquisição, temperatura de atuação que pode variar entre -60°C a 75°C, tensão máxima de 150VDC e potência máxima de 100mW (SUNROM TECHNOLOGIES, 2008, tradução nossa). Pelo fato dos valores de alimentação, não serem compatíveis com o microcontrolador utilizado, fez-se necessário a utilização de um resistor de 10kOhm no projeto.

O LDR, é composto por dois terminais, sendo um deles o GND (terra) e o outro referente ao VCC (alimentação) e a saída de dados analógica, sendo que este último deve estar intermediado pelo resistor na parte da alimentação.

Os dados coletados por meio da conexão com a porta analógica, são fornecidos em formato numérico, ao qual podem variar na faixa de 0 a 1010, sendo que, quanto mais próximo de zero é o valor obtido, maior é o valor da luminosidade

incidindo sobre o equipamento. Caso contrário, quanto maior o resultado, menor é a quantidade de luz que o ambiente possui.

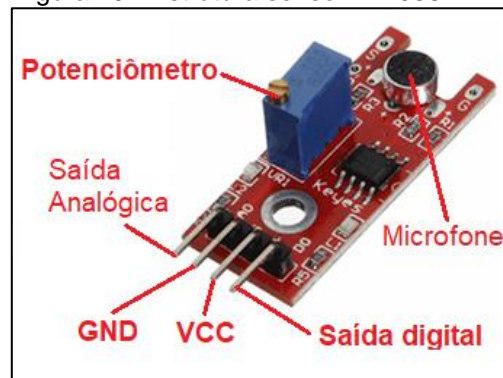
Já para a coleta das informações de sonoridade, utilizou-se do sensor KY-038, tendo este a proposta de realizar a detecção e mensurar os níveis de variações sonoras em um determinado ambiente, utilizando-se de um microfone condensador elétrico (MASTERWALKER, 2018).

Este dispositivo é formado por quatro terminais, sendo eles: alimentação, terra, saída analógica e saída digital, como também, possui um potenciômetro para o controle manual da sensibilidade de detecção do som e dois LEDs para a indicação de tensão e da saída digital.

Após a realização das conexões necessárias do sensor com o microcontrolador, os valores providos pela porta analógica, serão em formato numérico, representando a variação do som que está ocorrendo no ambiente. Já referente a porta digital, os valores obtidos variam de acordo com as detecções sonoras realizadas, onde para cada identificação efetuada, o equipamento alterna o estado de sua saída digital entre 0 e 1 (MASTERWALKER, 2018).

A figura 29 ilustra todas as ferramentas que compõem o sensor KY-038.

Figura 29 - Estrutura sensor KY-038



Fonte: Adaptado de Datasheetcafe (2018).

Dentre os principais motivos pelos quais optou-se pela aquisição e uso dos sensores listados anteriormente, se deram pelo baixo custo e pelas características e especificações que cada um contém, onde em sua maioria atendem a necessidades impostas no projeto.

A fim de realizar o controle e centralizar todos os dados coletados por cada um dos sensores citados anteriormente, utilizou-se do microcontrolador Arduino, em específico o modelo Mega 2560.

De acordo com o site oficial da plataforma (2019, tradução nossa), o Arduino Mega 2560 é um microcontrolador baseado na placa ATmega2560. Possui 54 pinos digitais, sendo que 15 destes podem ser utilizados como saídas PWM, 4 UARTs (portas seriais de hardware), 16 entradas analógicas, uma conexão USB, um oscilador de cristal de 16 MHz, um conector ICSP, um conector de alimentação e um botão reset.

Cada um dos 54 pinos digitais que o Mega contém, podem ser utilizados tanto como entrada, como para saída, podendo-se utilizar as funções *pinMode()*, *digitalWrite()* e *digitalRead()*. Sendo que, todas estas entradas operam com uma tensão de 5V, podendo fornecer e receber 20mA como condição de operação recomendada (ARDUINO, 2019, tradução nossa).

Este modelo pode operar a uma tensão de entrada variando de 7 a 12V, podendo ser alimentado por meio de uma conexão USB com um computador, ou diretamente em uma rede 220V, intermediada por uma fonte de 9V, por exemplo (ARDUINO, 2019, tradução nossa).

Ainda com relação as características de tensão de operação deste equipamento, o mesmo possui uma capacidade interna de proteção contra curtos e sobrecargas, onde para os casos em que a alimentação empregada seja superior a 500mA, existe um fusível que realiza o trabalho de contenção desta carga, até que a mesma seja normalizada.

Com relação ao armazenamento suportado por este modelo, o mesmo possui uma memória flash de 256KB, sendo que 8KB destes são usados no *bootloader*, possui também 8KB de SRAM e 4 KB de EEPROM. Já com respeito a velocidade de memória, possui 16MHz de Clock (ARDUINO, 2019, tradução nossa).

Além das entradas analógicas e digitais anteriormente citadas, este microcontrolador também possui uma área onde estão situados os pinos de alimentação, que possuem a finalidade de fornecer carga para conexões com equipamentos externos, dentre os pinos mais utilizados desta área, estão (SOUZA, 2014):

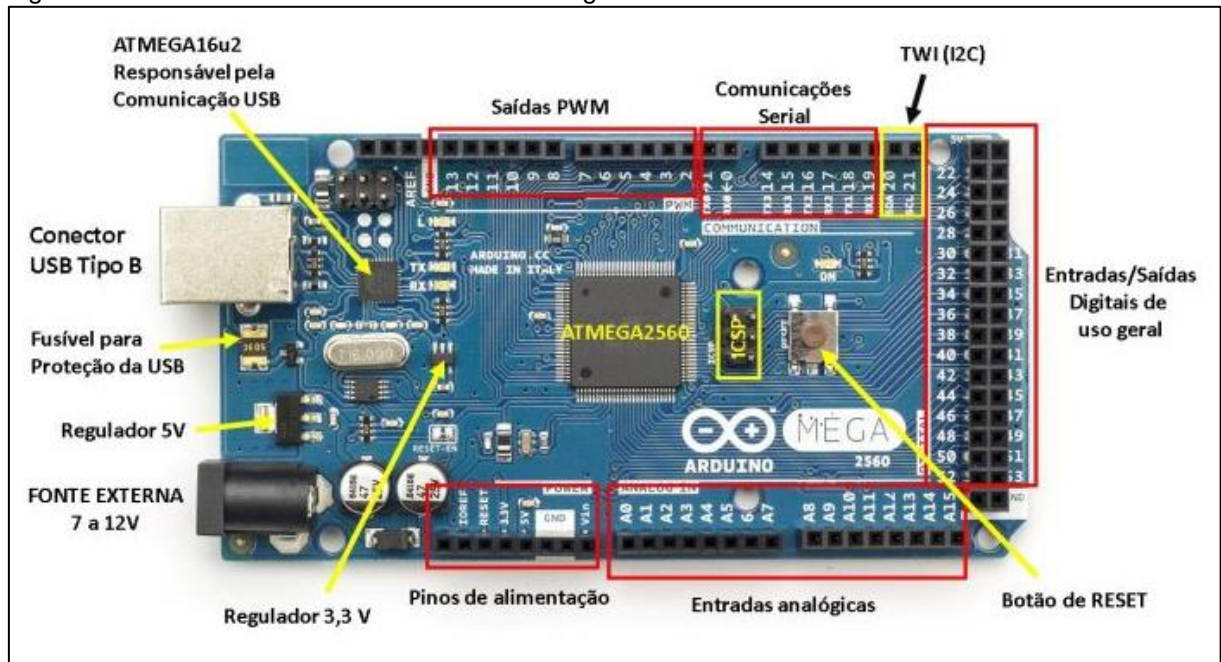
- a) 5V: disponibiliza tensão de 5V para a alimentação de dispositivos externos;

b) 3.3V: fornece alimentação diferenciada de 3.3V para equipamentos externos, tendo um consumo máximo de corrente de 50mA;

c) GND: terra.

A figura 30 ilustra todo o conjunto de componentes e recursos que compõem o Arduino Mega 2560, anteriormente citados.

Figura 30 - Resumo dos recursos do Arduino Mega2560



Fonte: Souza (2014).

Dentre as principais vantagens do Arduino, que nortearam a escolha pela sua utilização no projeto, estão no baixo custo de aquisição se comparado com outras plataformas, software de código aberto e extensível, ambiente de programação de fácil acesso e uso, multiplataforma, ao qual pode ser executado com Windows, Macintosh OS X e o Linux e principalmente pelo fato de que as aplicabilidades do Arduino são basicamente ilimitadas, dependendo apenas do empenho e criatividade aplicada pelos desenvolvedores.

Já com relação a escolha pelo Arduino Mega 2560, dentre todos os modelos que a família de microcontroladores Arduino contém, se deu principalmente pela sua maior robustez, visto que possui uma quantidade de entradas e saídas maior, com isso facilitando sua empregabilidade em projetos que requerem uma quantidade maior de conexões.

Após o término do acoplamento do microcontrolador e dos sensores responsáveis pela coleta dos dados, a próxima etapa constitui-se na utilização e nas configurações necessárias do equipamento responsável pelo envio das informações coletadas para o servidor na nuvem, ao qual para esta função, foi escolhido o módulo Wi-Fi ESP8266 ESP-01.

De acordo com as informações providas pelo *datasheet* fornecido pela Espressif Systems (2013, tradução nossa), este componente tem como seu principal objetivo, prover conexão com a Internet via Wi-Fi de forma eficaz e de baixo custo.

Possui capacidade de operar nas redes 802.11 b/g/n, ao qual pode atuar como um ponto de acesso ou no formato de uma estação, enviando e recebendo informação a uma distância que pode chegar em torno de 90m de acordo com cada ambiente (ESPRESSIF SYSTEMS, 2013, tradução nossa).

Sua tensão de operação é de 3.3V, suportando as comunicações do tipo TCP (do inglês, *transmission control protocol*) e UDP (do inglês, *user datagram protocol*), com até cinco conexões simultâneas, tendo a sua comunicação com o microcontrolador Arduino no formato serial, utilizando-se dos pinos TX e RX por meio de comandos AT (ESPRESSIF SYSTEMS, 2013, tradução nossa).

Com relação aos componentes que o mesmo possui, podem ser citados a antena de conexão embutida internamente, um led vermelho indicador de funcionamento, um led azul para indicação de atividade e mais oito pinos, que são (MORETTI, 2015):

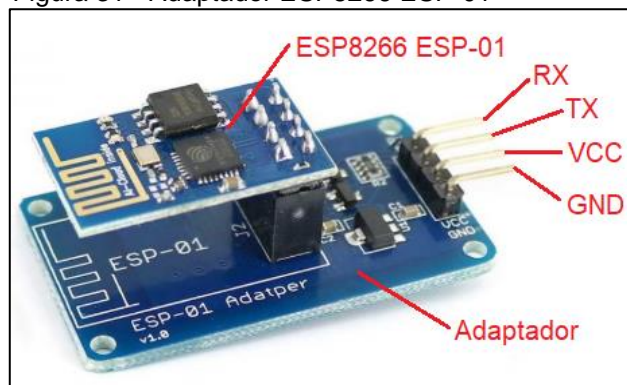
- a) VCC: tensão de alimentação de 3.3V, consumindo até 300mA;
- b) GND: terra;
- c) TX: sinal TX do módulo, deve ser conectado no RX do microcontrolador;
- d) RX: sinal RX do módulo, deve ser conectado no TX do microcontrolador;
- e) RST: sinal de *reset/restart*, é acionado em nível baixo, servindo para ocasiões em que se deseja reiniciar o módulo;
- f) CH_PD: é usado para situações em que se deseja realizar uma atualização de firmware do módulo, devendo ser usado em nível alto quando for operar normalmente;
- g) GPIO0: é controlado diretamente pelo firmware, devendo estar em nível baixo quando for realizado um update e em nível alto quando operado normalmente;
- h) GPIO2: I/O ao qual pode ser controlada diretamente pelo firmware;

Pelo fato do módulo ESP8266 ESP-01 não possuir uma tensão de entrada compatível diretamente com o módulo Arduino e também possuir uma alta quantidade de pinos, dificulta o manuseio e integração direta com o microcontrolador.

Como alternativa para contornar estes problemas, utilizou-se no projeto um adaptador para o módulo, ao qual este já realiza automaticamente a conversão da tensão de 5V do Arduino, para a tensão de operação do módulo Wi-Fi de 3.3V, como também diminui a quantidade de pinos para conexão direta com o microcontrolador.

Após o acoplamento do ESP8266 ESP01 a este adaptador, o mesmo passa a ter apenas 4 terminais, sendo eles: TX e RX, utilizados para transmissão dos dados e o VCC e GND relacionados a alimentação do módulo, com isso, agilizando o processo de interconexão entre o módulo e o microcontrolador. A figura 31 ilustra esta estrutura.

Figura 31 - Adaptador ESP8266 ESP-01



Fonte: Adaptado de StackExchange (2018).

Dentre as principais vantagens pelo qual escolheu-se a utilização da comunicação via Wi-Fi para a transmissão dos dados no projeto, estão na possibilidade de realizar o monitoramento das informações em tempo real e sem a necessidade de estar presente no local em que os sensores estão implantados.

Outra vantagem no uso deste formato, está na confiabilidade e segurança no armazenamento dos dados, visto que, os mesmos estão situados externamente em um servidor na nuvem, com isso, não estando acessíveis a qualquer pessoa e não ficando propensos a qualquer tipo de dano físico, se comparado ao armazenamento local em uma granja de produção de suínos.

Além dos benefícios já citados, o uso do Wi-Fi também proporciona a possibilidade de armazenamento dos históricos de coleta por muito mais tempo,

devido a diferença considerável no espaço que o servidor na nuvem pode conter, em comparação a um armazenamento local efetuado pelo microcontrolador, ao qual, neste último, seria necessário a utilização de um módulo de cartão SD por exemplo, podendo aumentar os custos e diminuindo a performance do projeto, visto que, seria necessária a extração dos dados através de uma conexão local, como o Bluetooth.

Com o término do acoplamento e das devidas conexões entre os dispositivos responsáveis pela coleta e envio dos dados para o servidor na nuvem, a última etapa da montagem da simulação, constitui-se na utilização e configurações dos equipamentos incumbidos de auxiliar no controle das variáveis de bem-estar animal.

A primeira parte desta etapa, deu-se na instalação de uma lâmpada, ao qual terá a função de simular o aquecimento do ambiente. Para isso, foi utilizado uma lâmpada incandescente de 40W de potência e tensão de 220V, ao qual foi acoplada a um soquete fixado ao suporte da simulação. E através deste, a mesma recebe a alimentação necessária para seu funcionamento de 220V.

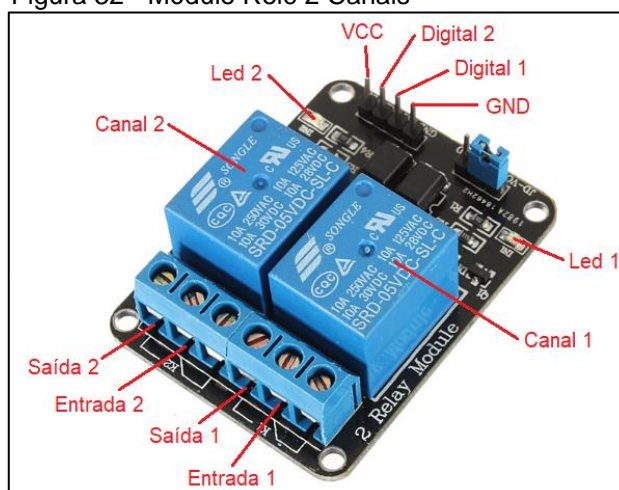
Posteriormente, foram instalados dois coolers com o intuito de simular o resfriamento do ambiente, tendo estes as tensões de operação de 5V cada, com isso, podendo ser conectados diretamente na alimentação provida pelo Arduino.

Após os equipamentos estarem devidamente instalados, foi implantado no projeto um módulo relé de 2 canais, a fim de realizar o controle do acionamento dos mesmos, por meio do gerenciamento da passagem de energia, ao qual pode ser manipulada diretamente pelo microcontrolador. O modelo em específico utilizado foi o *SRD-05VDC-SL-C*.

De acordo com as informações disponibilizadas no datasheet pela Songle Relay (2016, tradução nossa), esse modelo consegue realizar o acionamento de cargas de até 220VAC, trabalha em uma tensão de operação de 5VDC e possui um tempo de resposta variando de 5 a 10ms.

Com relação aos componentes que formam este equipamento (figura 32), são: dois canais de controle, onde cada um deles possui um led indicador de status e três entradas (normal aberto, normal fechado e comum). Também possui quatro terminais de comunicação com o microcontrolador, que são o VCC e GND relacionados a alimentação do módulo, e duas entradas digitais que controlam o chaveamento de cada um dos canais.

Figura 32 - Módulo Relé 2 Canais



Fonte: Adaptado de Songle Relay (2016).

Para a alimentação de todo o circuito montado, utilizou-se de uma fonte 9V para o microcontrolador, ao qual este redistribui a alimentação necessária para os demais componentes, com exceção da lâmpada, que da mesma forma que a fonte do Arduino, foi conecta diretamente a uma rede de energia de 220V.

Já para a realizar a conexão entre o Arduino e os demais componentes que são acoplados a ele, utilizou-se fios de cabo de rede do tipo par trançado, tendo estes a vantagem com relação aos jumpers, o fato de serem facilmente modulados de acordo com cada necessidade.

Com a montagem da simulação concluída, a próxima fase constituiu-se na programação para o microcontrolador, sendo esta, a responsável em fazer com que o Arduino realize todas as operações estipuladas para o projeto.

7.1.5 Desenvolvimento para o Arduino

A fim de realizar o desenvolvimento para o microcontrolador Arduino, utilizou-se da própria IDE disponibilizada no site oficial do equipamento, onde a partir desta, torna-se possível o desenvolvimento de programas, realização de importações e instalações de bibliotecas adicionais, compilação e carregamentos de programas para o Arduino.

A linguagem de programação utilizada para o desenvolvimento nesta IDE, baseia-se em C e C++, ao qual possui uma estrutura padrão contendo as funções

setup() e *loop()*, onde cada uma delas possuem as seguintes definições e funcionalidades:

- a) *setup*: possui como principal objetivo a realização das configurações iniciais do programa, como mensagens para o usuário, atribuição de valores as variáveis e definições para cada porta digital utilizada, indicando se a mesma será usada como entrada ou saída, entre outros. Tudo isso devido ao fato de que esta função é executada apenas uma vez no início do programa;
- b) *loop*: contém uma finalidade distinta da função *setup()*, onde esta deve ser composta por todas as operações que serão realizadas pelo Arduino, onde será executada repetidamente até que um sinal de pare seja enviado ao mesmo.

A figura 33 ilustra a estrutura básica de um programa para o Arduino.

Figura 33 - Estrutura básica de um programa para Arduino

```
#include "DHT.h"

String valor_analogico = "";
String valor_digital = "";

void setup() {
    pinMode(10, OUTPUT);
    pinMode(7, INPUT);
    pinMode(A6, INPUT);
}

void loop() {
    digitalWrite(10, HIGH);
    delay(1000);
    digitalWrite(10, LOW);
    delay(1000);
    valor_analogico = analogRead(A6);
    valor_digital = digitalRead(7);
}
```

Fonte: Do autor.

A figura 33, além de representar as funções padrões de um programa para Arduino, também ilustra outras funcionalidades gerais, dentre elas: a importação de uma biblioteca para o código fonte, realizada pelo comando *#include*; definições de variáveis; função *pinMode()* que tem por finalidade definir se uma determinada porta

do microcontrolador será de entrada ou saída; função *digitalWrite()* que tem por objetivo mudar o estado de uma porta digital; função *delay()* utilizada para dar uma pausa da execução, de acordo com o parâmetro passado a ela em milissegundos e as funções *analogRead()* e *digitalRead()* que servem para obter valores das portas analógicas e digitais respectivamente.

A primeira etapa do desenvolvimento do programa utilizado no projeto, constitui-se na leitura dos valores obtidos por meio das portas digitais e analógicas, aos quais os sensores de luminosidade e som foram acoplados no microcontrolador, por meio das funções *analogRead()* e *digitalRead()* comentadas anteriormente.

Já para a obtenção dos dados de temperatura e umidade, utilizou-se da biblioteca *DHT.h*, estando esta, disponível para a importação diretamente na IDE do Arduino, onde faz-se necessário a definição de qual porta analógica foi conectado o sensor e qual o modelo que foi utilizado, com isso, sendo possível a captura das informações de temperatura e umidade por meio das funções *readTemperature()* e *readHumidity()* respectivamente. A figura 34 ilustra a programação utilizada.

Figura 34 - Utilização da biblioteca DHT.h

```
#include "DHT.h"

String umidade = "";
String temperatura = "";

#define DHTPIN A7
#define DHTTYPE DHT11

DHT dht(DHTPIN, DHTTYPE);

void setup() {
    dht.begin();
}

void loop(void) {
    umidade = dht.readHumidity();
    temperatura = dht.readTemperature();
}
```

Fonte: Do autor.

Com a coleta dos dados concluída, o próximo passo realizado foi referente a programação voltada para a utilização do módulo Wi-Fi, ao qual será responsável pelo envio dessas informações para o servidor na nuvem.

Para isso, foram utilizadas duas bibliotecas, a *SoftwareSerial.h* disponível para importação na própria IDE do Arduino e a *ESP8266.h* disponível para download gratuito no GitHub⁸, onde também contém o manual de utilização.

A biblioteca *SoftwareSerial.h* possui como objetivo, replicar a possibilidade de comunicação serial em todas as portas digitais do Arduino, ao qual possui por padrão essa capacidade apenas nos pinos 0 e 1 (ARDUINO, 2019, tradução nossa). Já a *ESP8266.h*, tem como finalidade a disponibilização de uma série de funções para a utilização de todos os recursos disponíveis no módulo Wi-Fi *ESP8266 ESP-01*.

A figura 35 ilustra a programação necessária para a utilização do módulo Wi-Fi, a fim de realizar a comunicação com o servidor.

Figura 35 - Programação para o módulo Wi-Fi

```
#include <ESP8266.h>
#include <SoftwareSerial.h>

SoftwareSerial mySerial(52, 53); /*RX,TX*/
ESP8266 wifi(mySerial);
String dados = ""; /*Dados a serem enviados ao servidor*/

void setup(){
  if (wifi.joinAP(nome_wifi, senha_wifi))
    Serial.print("Conectado com sucesso, IP: " + wifi.getLocalIP().c_str());
  else
    Serial.println("Falha na Conexao Wifi");
}

void loop(void){
  wifi.createTCP(server, porta);
  String conteudo = "POST /conexao.php HTTP/1.0\r\n" +
    "Host: joaopauloniehueswessler.000webhostapp.com\r\n" +
    "Accept: *" + "/" + "*" + "\r\n" +
    "Content-Length: " + dados.length() + "\r\n" +
    "Content-Type: application/x-www-form-urlencoded\r\n" +
    "\r\n" + dados + "\r\n";
  wifi.send((const uint8_t*) conteudo, strlen(conteudo));
  wifi.releaseTCP();
}
```

Fonte: Do autor.

De acordo com o desenvolvimento ilustrado na figura 35, a primeira parte realizada, foi a definição de quais portas digitais do microcontrolador representam as

⁸ Site: https://github.com/itead/ITEADLIB_Arduino_WeeESP8266

conexões RX e TX junto ao módulo Wi-Fi, ao qual nestas, utilizou-se a comunicação serial por meio da biblioteca *SoftwareSerial.h*.

Após isso, foram utilizadas as funções disponibilizadas pela biblioteca *ESP8266.h* para o acesso aos recursos do módulo Wi-Fi, tendo início na *joinAP()*, que tem por objetivo realizar a conexão com um roteador Wi-Fi, onde são passados por parâmetro o nome e senha de acesso ao mesmo e a *getLocalIP()* que retorna o IP local adquirido pelo módulo após a conclusão da comunicação.

Com a conexão com a Internet realizada, torna-se possível a comunicação com o servidor, tendo início na utilização da função *createTCP()*, ao qual possui a finalidade de realizar a abertura de uma comunicação do tipo TCP, ao endereço e porta de acesso do servidor que são passados como parâmetro para a mesma.

Posteriormente é definido o cabeçalho da comunicação a ser realizada com o servidor, onde neste são definidos alguns parâmetros, como também o próprio conjunto de informações a ser enviado.

Dentre os parâmetros que são especificados nesta URL, podem ser citados: tipo da conexão HTTP a ser efetuada, ao qual no projeto utilizou-se do método POST; o endereço principal de acesso ao servidor; o endereço do recurso que será acessado dentro do servidor, neste caso o arquivo *conexao.php* e o conteúdo de informações que deseja ser transmitido, que são os dados obtidos através da coleta dos sensores.

Com relação ao conjunto de dados coletados que são anexados junto a esta URL de conexão, são formados por um arquivo no formato JSON, ao qual é contido pelo tipo da operação a ser identificado pelo PHP no servidor, os dados de coleta de cada sensor, sendo estes compostos por uma sigla para a identificação do sensor e os respectivos valores coletados e por fim, as chaves de acesso ao banco de dados. A figura 36 ilustra esta estrutura.

Para a concretização do envio das informações ao servidor, foi realizado o envio por parâmetro da URL configurada anteriormente, para a função *send()* e por fim utilizou-se da função *releaseTCP()* para o encerramento da conexão.

Figura 36 - JSON de dados enviados pelo Arduino

```

{
  "tipo": "1",
  "coleta": [
    {
      "sigla": "DHTEMP",
      "valor": "30"
    },
    {
      "sigla": "DHTUMI",
      "valor": "70"
    },
    {
      "sigla": "LDR",
      "valor": "500"
    },
    {
      "sigla": "KY038",
      "valor": "650"
    }
  ],
  "conexao": [
    {
      "user": "id_usuario",
      "senha": "XXX",
      "banco": "id_banco"
    }
  ]
}

```

Fonte: Do autor.

Por fim, a última etapa desenvolvida, constitui-se na programação para o controle do módulo relé, que possui o objetivo de gerenciar o acionamento dos equipamentos que compõe a simulação, e também, o desenvolvimento para controle do intervalo de tempo de verificação que deve ser realizado pelo microcontrolador.

A cada comunicação realizada com o servidor, efetuando o envio dos dados coletados, o retorno desta conexão fornecido pelo PHP é um arquivo no formato JSON, onde neste estão contidas as seguintes informações: status do envio realizado, indicando se houve sucesso ou erro ao gravar no banco de dados; status dos equipamentos, composto por uma sigla que o identifica e a última operação realizada pelo usuário na aplicação referente ao ligamento ou desligamento dos mesmos e o intervalo de tempo de leitura que o microcontrolador deve obedecer, também configurado pelo usuário na aplicação.

Com esse último status de cada equipamento, torna-se possível o controle de cada um dos canais do relé, por meio das suas respectivas portas digitais utilizando-se da função *digitalWrite()* comentada anteriormente, onde pode-se definir o valor de ligado e desligado de acordo com o retorno obtido para cada equipamento.

Já para o controle do tempo de verificação, utilizou-se da função *delay()*, passando para a mesma o tempo configurado pelo usuário, obtido através do JSON de retorno do PHP.

Após a conclusão de toda a programação voltada para o Arduino, tornando-se possível a realizar de toda a coleta e envio dos dados para o servidor na nuvem, a próxima fase do projeto refere-se ao desenvolvimento da aplicação para o gerenciamento dessas informações.

7.1.6 Desenvolvimento da aplicação

Após o término de toda configuração do ambiente e montagem da simulação, deu-se início ao desenvolvimento da aplicação para o monitoramento das informações utilizando-se do *framework* Ionic.

Dentre os motivos pelos quais escolheu-se o Ionic para o desenvolvimento, estão na possibilidade de criar aplicações móveis utilizando-se dos recursos de tecnologias web, como *HTML*, *CSS* e *JavaScript*, tendo como resultado uma aplicação que se comporta como nativa, ao qual pode desfrutar de todos os recursos do dispositivo.

Outro fator relevante para esta escolha, se dá pelo fato do Ionic possibilitar o desenvolvimento híbrido ou multiplataforma, ao qual, reduz consideravelmente o tempo de produção da aplicação, visto que, não se faz necessário ter conhecimentos específicos voltados para determinado sistema operacional.

Já com relação ao *framework* de *front-end* integrado ao Ionic, utilizado no projeto, foi o Angular, sendo este, uma plataforma utilizada para a construção de interfaces de aplicações desenvolvida pela Google, ao qual utiliza-se de tecnologias como *HTML*, *CSS* e *JavaScript*.

Uma característica marcante do Angular, está no fato dele ser de *open source*, possuindo uma grande comunidade ativa de desenvolvedores e empresas que o utilizam, com isso facilitando o ingresso de novos entusiastas, devido à grande quantidade de material disponível referente a ferramenta.

Para a criação de aplicações com Angular, pode-se usar as linguagens *JavaScript*, *TypeScript* e *Dart*. Porém como o Angular foi desenvolvido em *TypeScript*, a popularização desta, ficou maior junto ao framework em relação as demais. Em função disso, para o desenvolvimento deste projeto utilizou-se dele.

O *TypeScript* pode ser denominado como um superconjunto de *JavaScript*, contendo muito mais recursos, sintaxe simplificada e amplamente suportada pelos editores modernos. Também tendo como destaque, a disponibilidade de recursos para o uso da programação orientada a objetos. (DEV MEDIA, 2016).

Para dar início ao desenvolvimento, foram usados os recursos disponibilizados pelo Ionic CLI, possibilitando a criação de projetos via linha de comando, onde neste já é disponibilizado algumas opções de layouts pré-definidos, a fim de otimizar e dar maior produtividade ao desenvolvimento.

Para isso, utilizou-se do comando *ionic start*, seguido do nome do projeto e do *sidemenu*, que é referente ao nome do *layout* escolhido para dar início a aplicação, ao qual já disponibiliza um menu lateral pronto para a utilização. A figura 37 ilustra a utilização do comando.

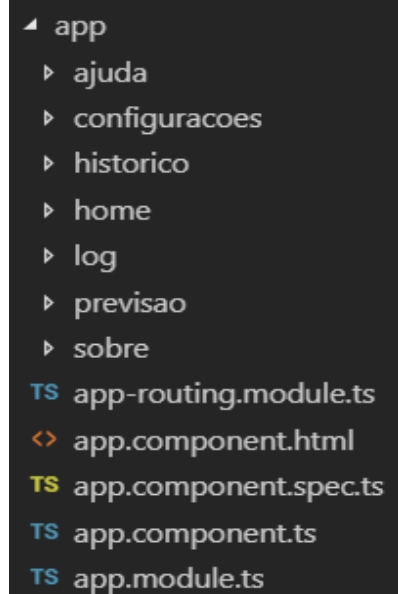
Figura 37 - Comando para criação de projetos em Ionic

```
C:\projetos> ionic start myApp sidemenu
```

Fonte: Do autor.

Após a criação do projeto, toda a estrutura de diretórios e arquivos necessários, já se encontram configurados, tendo como destaque, a pasta *app* (figura 38), visto que nesta é contido todo o pacote de códigos fontes desenvolvidos para a aplicação.

Figura 38 - Estrutura da pasta app



```
└─ app
  ├── ajuda
  ├── configuracoes
  ├── historico
  ├── home
  ├── log
  ├── previsao
  ├── sobre
  ├── app-routing.module.ts
  ├── app.component.html
  ├── app.component.spec.ts
  ├── app.component.ts
  └── app.module.ts
```

Fonte: Do autor.

Para a criação de cada uma das páginas da aplicação, pode-se realizar manualmente no diretório em questão, ou utilizar-se do comando *ionic generate page mypage*, disponibilizado pelo Ionic CLI, sendo o *mypage* o nome da página a ser criada.

Realizando esta operação via comando, o mesmo já cria automaticamente um diretório referente a página informada, e dentro deste, os arquivos *TypeScript* para criação das regras de negócio (*back-end*), o *HTML* para a interface com o usuário (*front-end*) e o *CSS* para a estilização.

Ainda com relação a estrutura da aplicação, outros dois arquivos que tem fundamental importância, são o *config.xml* e o *app-routing.module.ts*. Este primeiro tem como objetivo realizar algumas configurações de execução da aplicação, como por exemplo, o nome e o ícone que serão adicionados ao menu do dispositivo em que será instalado, como também uma animação para a tela inicial de carregamento da aplicação, sendo que, estas configurações podem ser realizadas especificamente para cada plataforma móvel, como Android e IOS.

Já com relação ao arquivo *app-routing.module.ts*, o mesmo tem como objetivo principal definir as rotas ou localização de cada página dentro da aplicação. Conforme pode ser observado na figura 39, o atributo *path* representa o nome da página e o *loadChildren* armazena o seu respectivo endereço dentro do diretório do projeto. Com isso, a cada navegação realizada dentro do aplicativo, é passado via parâmetro qual o nome do destino, e desta forma, o Ionic automaticamente redireciona para o caminho configurado.

Figura 39 - Arquivo *app-routing.module.ts*

```
import { NgModule } from '@angular/core';
import { Routes, RouterModule } from '@angular/router';

const routes: Routes = [
  { path: '', redirectTo: 'home', pathMatch: 'full' },
  { path: 'home', loadChildren: './home/home.module#HomePageModule' },
  { path: 'sobre', loadChildren: './sobre/sobre.module#SobrePageModule' },
  { path: 'ajuda', loadChildren: './ajuda/ajuda.module#AjudaPageModule' },
  { path: 'historico', loadChildren: './historico/historico.module#HistoricoPageModule' },
  { path: 'previsao', loadChildren: './previsao/previsao.module#PrevisaoPageModule' },
  { path: 'log', loadChildren: './log/log.module#LogPageModule' }
];

@NgModule({
  imports: [RouterModule.forRoot(routes)],
  exports: [RouterModule]
})
export class AppRoutingModule {}
```

Fonte: Do autor.

Para a programação do *front-end* da aplicação, manteve-se um padrão para todas as telas criadas, contendo uma estrutura básica no arquivo *HTML*, formadas pelo cabeçalho (*ion-header*), conteúdo (*ion-content*) e rodapé (*ion-footer*), onde para cada página, essas *tags* contém componentes específicos a fim de atender às necessidades individuais de cada uma delas. A figura 40 representa esta estrutura.

Figura 40 - Estrutura arquivo HTML

```
<ion-header>
|   Cabeçalho
</ion-header>

<ion-content padding>
|   <span>Conteúdo</span>
|   <ion-list>
|       <ion-item *ngFor="let log of logObj">
|           <span>{{log.valor}}</span>
|           <span>Data: {{log.data}} - {{log.hora}}</span>
|       </ion-item>
|   </ion-list>
</ion-content>

<ion-footer>
|   Rodapé
</ion-footer>
```

Fonte: Do autor.

Também pode ser observado na figura 40, alguns recursos para a integração da parte visual com o *back-end*, como por exemplo o *ngFor*, sendo este capaz de exibir em uma lista (*ion-list*) um *array* de objetos criado e carregado no código fonte, como também, a abertura e fechamento duplo de chaves, onde o conteúdo por estas envolvido, pode representar uma variável definida no código, fazendo com que automaticamente seu valor seja exibido em tela.

O *back-end* da aplicação, teve como linguagem de programação base o *TypeScript*, onde nesta etapa de desenvolvimento, um dos processos com maior relevância, refere-se a conexão da aplicação com o servidor na nuvem, através do PHP citado anteriormente, ao qual por meio deste, irá ocorrer toda a transmissão dos dados entre o aplicativo e o banco de dados.

A fim de realizar esta comunicação, utilizou-se de uma API do Angular, o *HttpClient*, sendo esta responsável pelo envio e captura de informações realizadas via protocolo HTTP entre clientes e servidores através da Internet (ANGULAR, 2019).

O primeiro passo para o desenvolvimento deste processo, foi a realização da importação desta API, conforme pode ser observado na figura 41.

Figura 41 - Importação API HttpClient

```
import { HttpClient } from '@angular/common/http';
```

Fonte: Do autor.

Após isso, implementou-se o uso da API no código fonte da aplicação, onde neste devem ser especificados qual o método HTTP utilizado e a URL que será realizada a conexão, e por meio desta deve ser passado os parâmetros necessários para que o serviço chamado entenda qual a solicitação desejada.

Na aplicação foram utilizados os métodos GET e SET de acordo com cada transmissão a ser realizada, a figura 42 ilustra a implementação de um GET para a obtenção de dados do PHP, onde é possível a manipulação destas informações através da variável *data*. Com isso, torna-se possível a atribuição desta variável a um *array* de objetos, como no exemplo o *logObj*, onde a partir disto torna-se possível a integração com o *front-end*, conforme exemplo ilustrado e explicado anteriormente na figura 40.

Figura 42 - Implementação do método GET

```
urlConex = 'https://joaopauloniehueswessler.000webhostapp.com/conexao.php' +
'?dados={"tipo":"2","conexao":[{"user":"id8898795_admin" +
',"senha":"XXX","banco":"id8898795_bancotcc"}]}';

this.http.get(urlConex).subscribe(
  (data : any) => {
    this.logObj = data;
  }, (error : any) => {
    alert("Erro de conexão");
  }
);
```

Fonte: Do autor.

Através deste formato de conexão entre aplicativo e servidor, possibilitou-se o desenvolvimento do mesmo de forma inteiramente *online*, ao qual faz-se necessário possuir comunicação com a Internet para realizar todas as operações

disponíveis em sua utilização, visto que, os dados que são exibidos e gravados a partir dele, são transmitidos diretamente do servidor em tempo real.

Na aplicação, além das interfaces para o usuário interagir com os dados coletados pelo microcontrolador, também foi disponibilizado uma opção para a visualização da previsão do tempo, ao qual tem o objetivo de auxiliar e prover uma melhor organização a pessoa responsável pelo manejo dos animais, visto que de certa forma, pode prever o clima para os dias seguintes.

Para a obtenção destes dados de previsão do tempo, utilizou-se de uma API⁹ gratuita disponibilizada por uma empresa privada de meteorologia, a Climatempo. A fim de obter o acesso, é necessário realizar a criação de uma conta, a partir disso, o usuário recebe uma chave de acesso e já pode realizar consultas as ferramentas disponibilizadas pela API, com um limite de 300 requisições diárias por chave de acesso para o plano gratuito.

Este processo de obtenção dos dados, é subdividido por cidades. Para isso, existe um serviço específico apenas para a identificação destas, tendo a possibilidade de busca pelo nome e estado ou pela localização do dispositivo. Feito isso, este serviço retorna o código da cidade em questão e com isso, é possível utilizá-lo na API principal a fim de obter as informações de previsão do tempo. Para o desenvolvimento da aplicação, foi utilizado o código fixo da cidade de Criciúma.

Como forma de conexão entre a aplicação e a API, utilizou-se do mesmo formato usado para a comunicação com o PHP anteriormente citado, através de requisições HTTP utilizando-se o método GET. Com relação aos dados obtidos, os mesmos são providos em formato JSON, onde são tratados pelo aplicativo e exibidos para o usuário.

Após o desenvolvimento estar concluído, faz-se necessário adicionar a plataforma do sistema operacional móvel que se deseja utilizar no diretório do projeto em questão, como por exemplo o Android. Para isso, deve ser realizado o comando *ionic platform add android*.

Por fim, para realizar os testes da aplicação, dispõe-se de duas possibilidades, a primeira delas é executar a aplicação diretamente através do navegador de Internet com o comando *ionic serve*. Já a segunda, refere-se a compilar o aplicativo para rodar diretamente em um *smarthphone*, podendo ser realizada por

⁹ Site: <https://advisor.climatempo.com.br>

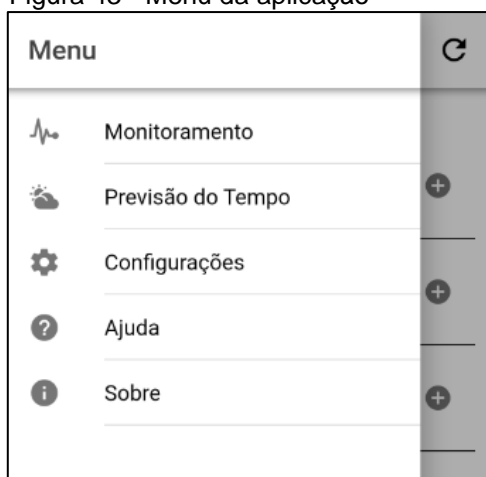
meio do comando *ionic cordova run android* em dispositivos com Android, desde que o aparelho esteja devidamente conectado e liberado.

A fim de manter as boas práticas de programação e versionamento de código, utilizou-se do GitHub¹⁰ para o armazenamento do projeto em questão, sendo este, um serviço online de hospedagem de repositórios, ao qual possui uma popularização mundialmente reconhecida entre os desenvolvedores, fazendo com que a maior parte dos projetos, frameworks e bibliotecas que estão surgindo de caráter de código aberto, sejam nele armazenados.

7.1.8 Funcionamento da aplicação

Após o término da etapa de desenvolvimento da aplicação, a mesma ficou composta por um conjunto de páginas, cada qual com suas finalidades específicas. Dentre estas, existem telas de consultas dos dados e de manutenção, sendo que todas elas são organizadas pelo conceito de menu (figura 43), onde a partir deste é possível acessar todos os recursos do aplicativo.

Figura 43 - Menu da aplicação



Fonte: Do autor.

Baseando-se em uma lógica para o acesso cronológico das telas na aplicação, tem-se na primeira página a de monitoramento (figura 44) sendo a principal delas, visto que é nesta que o usuário poderá acompanhar os últimos valores de cada

¹⁰ <https://github.com>

sensor obtido pelo microcontrolador, com sua respectiva data e hora, e se julgar necessário, poderá realizar o acionamento dos equipamentos disponíveis.

Figura 44 - Tela de monitoramento



Fonte: Do autor.

A partir desta tela também, torna-se possível realizar uma consulta detalhada (figura 45) de todas as leituras realizadas pelo microcontrolador, individualmente por sensor em um período de datas, como também, sendo possível tanto através da página de histórico como na principal, a visualização de problemas relacionados a estrapolação dos valores obtidos, com relação as faixas parametrizadas, visto que, o aplicativo de forma automática, destaca em cor vermelha para a melhor percepção do usuário.

Figura 45 - Tela de histórico de coleta



Fonte: Do autor.

Seguindo a ordem de apresentação das páginas dispostas no menu, tem-se a tela de previsão do tempo (figura 46). Nesta o usuário tem a possibilidade de visualizar a previsão do tempo para os próximos sete dias, sendo ilustrados as principais informações com relação ao clima, como: temperatura, umidade, sensação térmica, vento, precipitação, chance de chuva e uma breve descrição de como será o tempo na data em questão.

Figura 46 - Tela de previsão do tempo



Fonte: Do autor.

A próxima página, refere-se a configuração (figura 47), onde o usuário tem a possibilidade de parametrizar o intervalo de tempo de verificação do microcontrolador, como também as faixas de valores ideais para cada sensor, ao qual para os casos onde os valores coletados não estiverem compreendidos em suas respectivas faixas, a tela inicial de monitoramento irá demonstrar ao usuário em cor vermelha, para que o mesmo fique ciente do problema.

Figura 47 - Tela de configuração

The screenshot shows a mobile application interface for configuration. At the top, there is a header bar with a hamburger menu icon, the title 'Configurações', and two icons: a clipboard and a save icon. Below the header, the screen is divided into three main sections:

- Tempo de verificação em segundos:** A section containing a single input field with the value '30'.
- Limites de bem-estar por sensor:** A section containing a table of sensor limits.

	Inicial	Final
Temperatura:	23	35
Umidade:	60	80
Luz:	15	1000
Som:	8	905
- Automatização:** A section containing a green toggle switch that is currently turned on.

Fonte: Do autor.

Na tela de configuração, tem-se também a opção para a realização da automatização do acionamento dos equipamentos, onde está voltada diretamente para os valores de temperatura. Caso esta opção esteja ativa, o sistema irá automaticamente validar se a última temperatura obtida está dentro dos limites ideais parametrizados, caso contrário, se estiver abaixo da faixa inicial, a lâmpada será acesa simulando o aquecimento e se estiver acima da faixa final, os coolers serão ligados simulando o resfriamento.

Esse controle realizado pela opção de automatização, acontece toda vez que o microcontrolador faz o envio dos dados coletados para o servidor, onde lá, o PHP valida as informações enviadas, verificando se esta opção está ativa e se os dados estão fora da faixa parametrizada, com isso retornando o status de acionamento referente a cada equipamento que o Arduino deve realizar.

Outro recurso presente nesta etapa de configuração, está no fato de que o usuário tem a possibilidade de acompanhar todo o log de alterações (figura 48) realizadas no aplicativo, como por exemplo, nas modificações realizadas nas configurações ou nos acionamentos efetuados nos equipamentos.

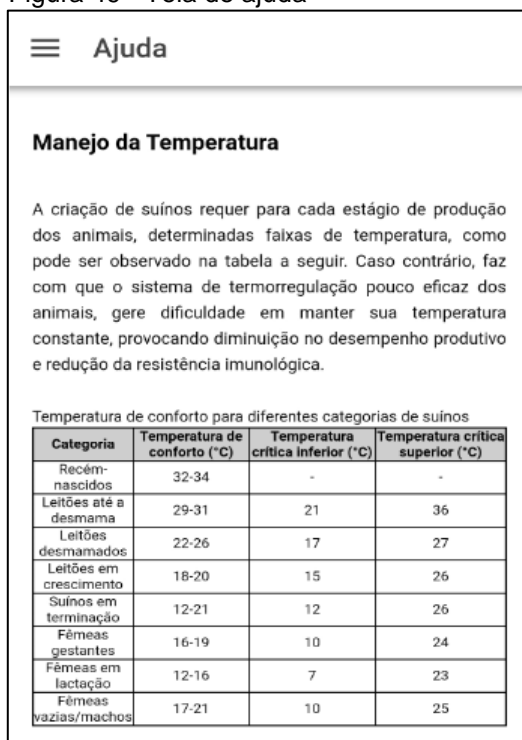
Figura 48 - Tela de logs de alteração



Fonte: Do autor.

A tela subsequente, trata-se de uma opção de ajuda (49) ao usuário, onde nela contém de forma sucinta a importância do manejo de cada uma das variáveis de bem-estar animal, bem como os níveis ideais para cada uma delas, como temperatura, umidade, sonoridade e luminosidade.

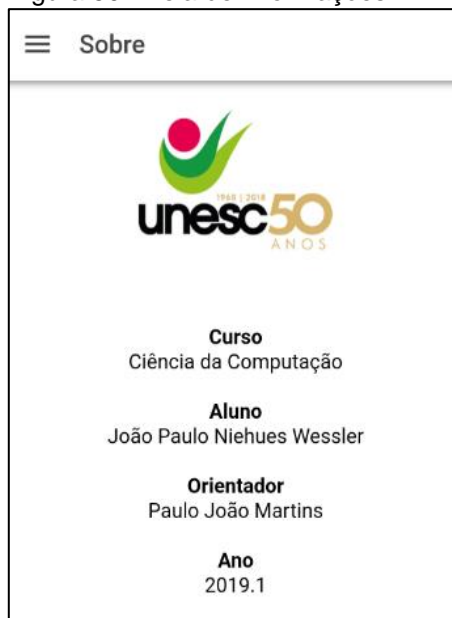
Figura 49 - Tela de ajuda



Fonte: Do autor.

Por fim, uma página de informações do aplicativo (figura 50), contendo os dados dos responsáveis pelo projeto e da universidade.

Figura 50 - Tela de informações



Fonte: Do autor.

7.1.9 Testes do Projeto

Após a conclusão de todas as fases de desenvolvimento do projeto, deu-se início a realização dos testes, ao qual teve como primeira etapa, a validação da simulação criada.

Para isso, iniciou-se pela configuração da base de dados inicial necessária, onde é preciso ser criado manualmente no banco, os registros dos sensores e equipamentos utilizados em suas respectivas tabelas, tendo cada registro incluído, a necessidade de informar a sigla correspondente que foi especificada na programação para o Arduino. Com isso, a cada operação realizada, torna-se possível identificar qual o cadastro do sensor ou equipamento que está sendo operado através da mesma.

Com a base de dados devidamente configurada, torna-se possível ligar a simulação, afim de averiguar o funcionamento da coleta das informações e acionamento dos equipamentos.

Para verificação de acurácia dos dados coletados, foi colocado a maquete em três tipos de ambiente com acesso à rede Wi-Fi, sendo eles: ambiente fechado,

sem iluminação e úmido; ambiente aberto, com iluminação natural e ventilado e um ambiente fechado, com iluminação artificial e com pouca ventilação.

Em ambos os três ambientes, o microcontrolador conseguiu operar normalmente, realizando as coletas de dados e operando os acionamentos realizados em cada equipamento, sendo que neste último realizou as operações tanto com o modo de automatização ativado, como no formato manual.

Para os testes do funcionamento da opção de automatização, utilizou-se de um secador de cabelo. Com isso, conseguiu-se realizar alterações abruptas na temperatura em torno da simulação, onde para cada extrapolação identificada no valor obtido pelo sensor, com relação as faixas parametrizadas na aplicação, o microcontrolador realizou corretamente o acionamento dos equipamentos.

A figura 51 demonstra a efetivação dos dados de coleta enviados para o banco de dados no servidor na nuvem.

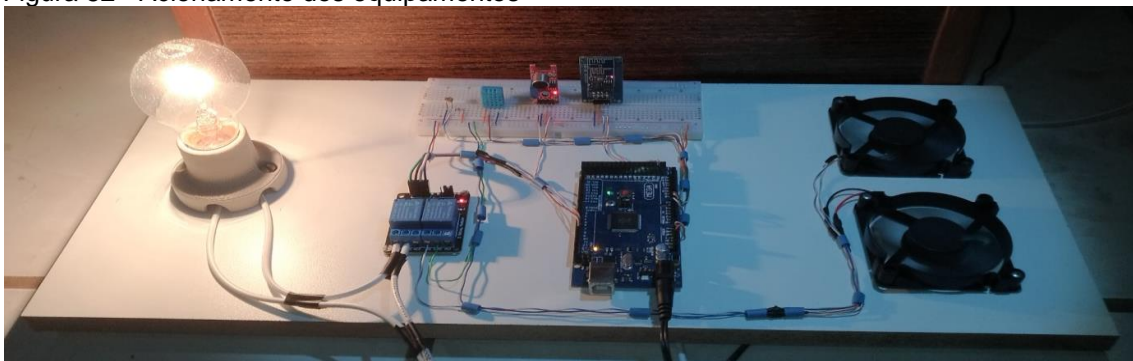
Figura 51 - Registros de coleta na base de dados

id_coleta	data	hora	valor	descricao
31026	2019-06-02	18:43:03	605	Som
31025	2019-06-02	18:43:03	989	Luz
31024	2019-06-02	18:43:03	92.00	Umidade
31023	2019-06-02	18:43:03	22.00	Temperatura
31022	2019-06-02	18:42:46	605	Som
31021	2019-06-02	18:42:46	989	Luz
31020	2019-06-02	18:42:46	92.00	Umidade
31019	2019-06-02	18:42:46	22.00	Temperatura
31018	2019-06-02	18:42:29	605	Som
31017	2019-06-02	18:42:28	990	Luz
31016	2019-06-02	18:42:28	92.00	Umidade
31015	2019-06-02	18:42:28	22.00	Temperatura

Fonte: Do autor.

A figura 52 ilustra o acionamento manual dos equipamentos na simulação.

Figura 52 - Acionamento dos equipamentos



Fonte: Do autor.

Os testes da aplicação, foi realizado através de um smartphone com sistema operacional Android na versão 8.0.0, o ZenFone 3 da marca Asus. Com isso, foi verificado a funcionabilidade geral do aplicativo, como: menu principal fornecendo acesso a todas as opções nele contidas; navegação entre telas e demais funcionalidades, tendo em todas um resultado final positivo, conforme ilustrado anteriormente nas figuras do tópico 7.1.8.

A execução da aplicação também no sistema operacional IOS, que é proporcionada pelo framework Ionic, não foi realizada no projeto, visto que, para este SO, a geração do aplicativo na ferramenta, faz-se necessário possuir o SDK da Apple, tendo este o funcionamento exclusivo apenas no seu próprio SO, ao qual não se teve acesso durante os testes do projeto.

7.2 RESULTADOS E DISCUSSÕES

Com o término de todas as etapas estipuladas para o projeto e com os testes realizados nas funcionalidades disponibilizadas por ele, tornou-se possível realizar uma análise relacionada aos seus resultados, com isso verificando a sua viabilidade e aplicação.

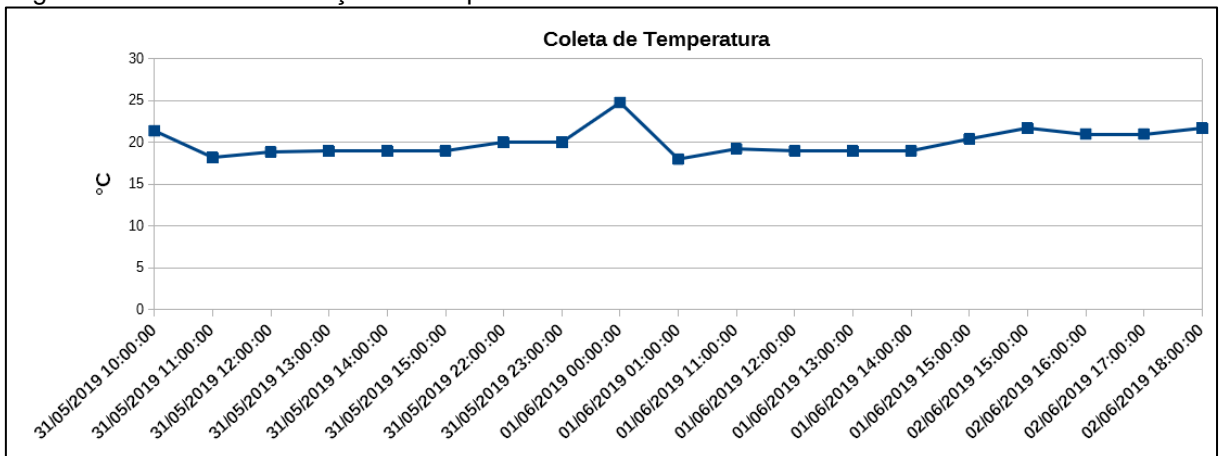
Relativo à simulação criada e os equipamentos nela utilizados, pode-se perceber que o microcontrolador Arduino, que possui o papel principal de gerenciamento dentro da simulação, teve uma boa integração com os demais módulos e sensores empregados, visto que, todos responderam e operaram de acordo com os comandos por ele repassado.

Já com relação a verificação da acurácia fornecida individualmente por cada sensor, foram extraídos da base de dados em formato de arquivo de texto, 1.501 registros coletados durante o período de três dias em que o circuito permaneceu alguns momentos ligado, possibilitando assim, a importação desses dados na ferramenta LibreOffice Calc, a fim de realizar a geração de gráficos que melhor auxiliam na visualização.

Para proporcionar uma melhor validação nos gráficos gerados, devido à grande quantidade de registros, os dados foram organizados entre intervalos de tempo de uma hora, ao qual aplicou-se uma média simples entre os mesmos.

A figura 53 ilustra o gráfico de variação dos valores coletados de temperatura a partir do sensor DHT11.

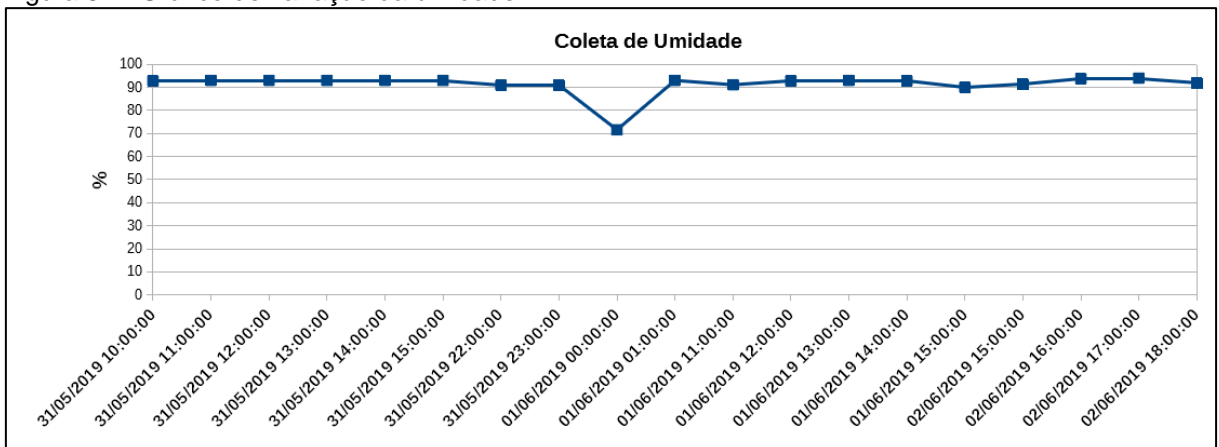
Figura 53 - Gráfico de variação de temperatura



Fonte: Do autor.

A variação de umidade também coletada pelo sensor DHT11, pode ser observada na figura 54.

Figura 54 - Gráfico de variação da umidade

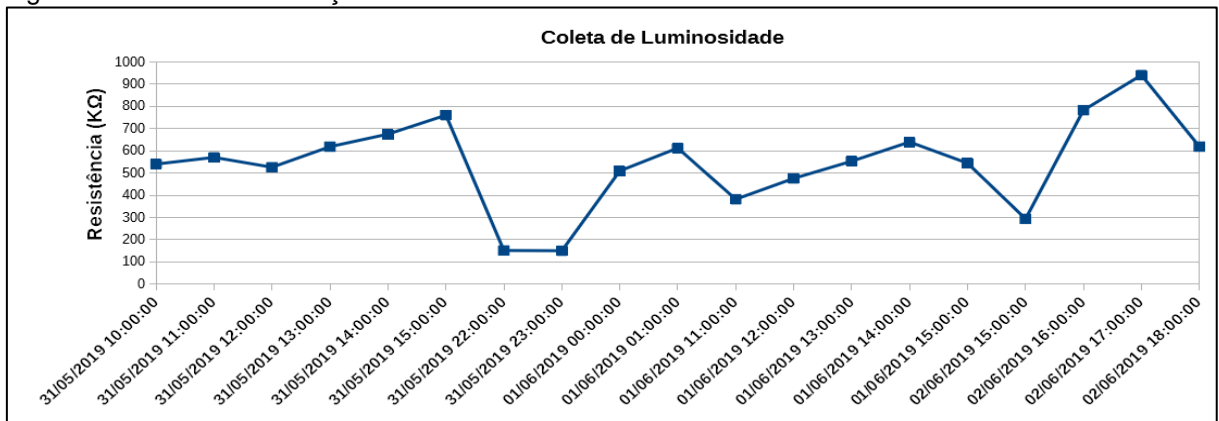


Fonte: Do autor.

Conforme ilustrado anteriormente em ambos os gráficos de temperatura e umidade, pode ser observado que no dia 01/06/2019 na hora 00:00 ocorreu alta variação, isso devido ao fato de que a simulação neste momento foi exposta a uma fonte de calor, proporcionando um aumentando na média da temperatura e diminuindo a média da umidade, representando o correto funcionamento do sensor.

A figura 55 ilustra o gráfico de variação relativo aos dados de coleta de luminosidade obtidos através do sensor LDR.

Figura 55 - Gráfico de variação de luminosidade

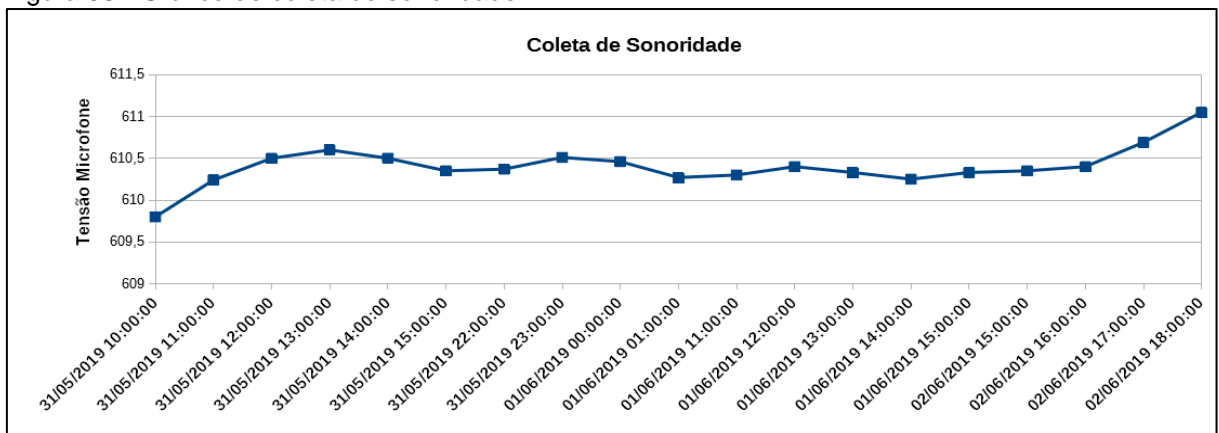


Fonte: Do autor.

De acordo com o gráfico ilustrado, há uma forte variação nos valores de luminosidade coletados, visto que, nos dias que foi realizada a extração dos dados, a simulação estava localizada em um ambiente exposto a iluminação artificial, com isso sofrendo constantes alterações. De toda forma, ilustrando o funcionamento correto do sensor.

Relativo às informações de coleta de sonoridade, o gráfico ilustrado na figura 56 representa a variação obtida através do sensor KY-038, onde mesmo utilizando-se dos métodos anteriormente citados no item 7.1.5, não se obteve êxito em extrair do sensor a variação sonora, independentemente da existência ou não de algum ruído no ambiente, ao qual os dados permaneceram constantes de acordo com a regulação realizada no potenciômetro do sensor.

Figura 56 - Gráfico de coleta de sonoridade



Fonte: Do autor.

Com exceção do sensor sonoro, que não se conseguiu obter resultados satisfatórios, os demais apresentaram bons resultados de acordo com as coletas realizadas, ao qual, mesmo não sendo sensores de carácter profissional com uma alta acuracidade, possuem um bom potencial se visado o seu baixo custo de aquisição.

Da mesma forma que os estudos realizados por Matos (2017) e Borges et al. (2018), a aplicação do microcontrolador Arduino integrado a sensores, mostrou-se uma forma eficiente de realizar verificações de variáveis de ambiente, mesmo com um baixo investimento, ao qual torna-se possível a automatização de processos, podendo facilitar e agilizar as tarefas onde o mesmo é empregado.

Já com respeito a confiabilidade no modelo de envio dos dados para o banco de dados na nuvem, realizado através do módulo Wi-Fi ESP8266 ESP-01, verificou-se que para uma ocasião onde o intervalo de verificação esteja parametrizado como 30 segundos na aplicação, o protótipo deveria realizar em torno de 2.160 coletas em um intervalo de 24 horas.

Com testes reais realizados referente a esta situação, em 24 horas foram obtidas 2.087 coletas devidamente efetuadas no banco de dados, chegando a um índice de 96.62% de eficácia, 1.38% a menos se comparado ao projeto realizado por Oliveira (2015) em sua pesquisa, onde o mesmo obteve 98% de precisão, tendo como principal motivo para a diferença, a queda de conexão com a Internet no ambiente em que foi realizado os testes. De toda forma, o protótipo teve uma perda pequena diante da quantidade de leituras realizadas.

Quanto ao servidor *000webhost*, utilizado para o armazenamento no projeto, mostrou-se eficiente nas atividades de que o mesmo foi incumbido, mantendo uma boa performance constantemente e não apresentando nenhum tipo de problema com relação a acesso ou utilização, mesmo sendo disponibilizado gratuitamente.

Relacionado ao desenvolvimento da aplicação utilizando-se do *framework* Ionic, da mesma maneira que Mazuchetti (2015), o uso deste facilita a linha de aprendizado e o próprio desenvolvimento, visto que este possibilita o uso de linguagens de programação web, para a produção de aplicativos com um resultado final similar a um nativo, o que se torna um grande diferencial além do benefício de ser multiplataforma e não requerer conhecimento voltado para as linguagens específicas de cada sistema operacional.

Ainda com relação a aplicação, os resultados obtidos com a mesma são satisfatórios, visto que se obteve um aplicativo capaz de realizar todas as operações

propostas no projeto, quanto ao gerenciamento das variáveis coletas, conforme mencionado no item 7.1.8.

Da mesma forma que Lee e Yeo (2010, tradução nossa) e Oliveira (2015), pode-se dizer que o uso da tecnologia para o monitoramento de variáveis em ambiente de produção animal, pode otimizar e proporcionar bons níveis de automatização, com isso realizando a redução de custos operacionais e consequentemente aumentando a produtividade dos mesmos.

8 CONCLUSÃO

Com o término do trabalho em questão, além proporcionar a aquisição de conhecimento relacionado as tecnologias utilizadas para a realização do protótipo, também forneceu noções básicas referente aos conceitos relacionados a produção de suínos, ao qual pode-se compreender a importância que o manejo realizado corretamente, referente as variáveis de monitoração selecionadas, tem como influência no resultado final da produtividade dos animais.

Através do uso do microcontrolador Arduino juntamente com os demais dispositivos acoplados, tornou-se possível realizar o monitoramento das variáveis de ambiente de temperatura, umidade, luminosidade e sonoridade, obtendo-se êxito na realização da coleta das informações e envio para o servidor na nuvem, tendo apenas como empecilho os dados obtidos de frequência sonora, onde não se mostraram atrativos para um real monitoramento.

Referente ao processo de desenvolvimento da aplicação, o mesmo possibilitou a aquisição de conhecimento à cerca do conceito de aplicações híbridas, realizadas por meio de *frameworks* multiplataformas, em específico o Ionic, onde foram elencadas as principais funcionalidades e benefícios que o mesmo disponibiliza, afim de ser implementado no projeto.

Com isso, foi possível obter como resultado, um aplicativo capaz de realizar os procedimentos propostos no projeto, utilizando-se de métodos ágeis e dinâmicos de programação fornecidos pelo Ionic, tendo estes, as características principais voltadas inteiramente para a utilização de tecnologias de desenvolvimento web.

Durante o desenvolvimento do projeto, algumas dificuldades foram encontradas com relação a parte prática, dentre elas podem ser citadas a complexidade na programação para o uso do módulo Wi-Fi, ao qual fez-se necessário extensas pesquisas e tentativas até que se descobriu a biblioteca correta a ser utilizada.

Outro empecilho enfrentado, foi relacionado ao sensor sonoro KY-038, apesar do mesmo não possuir um grau de dificuldade elevado, foram realizadas diversas formas de programação e testes, devido a não ter obtido êxito em sua utilização.

Relativo ao desenvolvimento da aplicação, o contratempo de maior relevância encontrado, se refere a dificuldade em encontrar material de apoio

relacionado a versão quatro do Ionic, visto que, a mesma teve seu lançamento oficial poucos meses antes do início da produção do aplicativo. Com isso, levando a utilização do método de tentativa e erro para determinadas funcionalidades utilizadas.

Com relação ao projeto desenvolvido como um todo e com base nos testes realizados, obteve-se um resultado satisfatório quanto a integração entre todas as tecnologias implementadas, tendo como desfecho um protótipo de sistema de monitoramento de variáveis de bem-estar animal, que pode ser avaliado para a aplicação em um ambiente de produção de suínos, afim de otimizar e gerar melhores índices de produtividade.

De acordo com o conhecimento obtido e nos resultados adquiridos com o projeto, sugere-se como trabalhos futuros:

- a) aplicar o protótipo em um cenário de produção de suínos, afim de se obter dados reais, possibilitando uma melhor análise e percepção de possíveis melhorias no fluxo estipulado;
- b) implementar o uso de tecnologias que possibilitam a comunicação entre microcontroladores, como por exemplo o ZigBee, a fim de conseguir expandir a área de monitoramento;
- c) aplicar o conceito de acesso por usuários e perfis de usuários na aplicação e na base de dados, provendo segurança e privacidade das informações, visto que no formato atual, todos os dispositivos com o aplicativo instalado, estão acessando o mesmo conjunto de informações na base de dados;
- d) avaliar a possibilidade de implementação de algum modelo de mineração de dados, ao qual pode prover o uso de inteligência artificial para a automatização dos equipamentos;
- e) verificar outras variáveis de monitoração relacionadas ao bem-estar animal na suinocultura e que sejam passíveis de implementação no projeto;
- f) avaliar a implementação de uma tecnologia de comunicação local, a fim de atender a demanda de ambientes que não possuem acesso externo à Internet.

REFERÊNCIAS

ABNT. **Normalização**. Disponível em: <http://www.abnt.org.br>. Acesso em: 24 nov. 2018.

ANDROID. **Arquitetura da plataforma**. 2019. Disponível em: <https://developer.android.com/guide/platform/?hl=pt-br#linux-kernel>. Acesso em: 05 fev. 2019.

ANGULAR. **HttpClient**. 2019. Disponível em: <https://angular.io/guide/http>. Acesso em: 01 jun. 2019.

AOSONG ELECTRONICS. **Digital-output relative humidity & temperature sensor/module DHT22 (DHT22 also named as AM2302)**. Disponível em: <https://www.sparkfun.com/datasheets/Sensors/Temperature/DHT22.pdf>. Acesso em: 05 out. 2018.

_____. **Digital relative humidity & temperature sensor DHT11**. 2019. Disponível em: https://www.electronicoscaldas.com/datasheet/DHT11_Aosong.pdf. Acesso em: 29 jan. 2019.

_____. **Humidity sensitive resistor Product Manual HR202L**. Disponível em: <https://akizukidenshi.com/download/ds/aosong/HR202L.pdf>. Acesso em: 08 out. 2018.

_____. **Resistive humidity sensor, Model: HR202L**. Disponível em: <https://opencircuit.nl/ProductInfo/1000094/HR202L.pdf>. Acesso em: 08 out. 2018.

APPPFIGURES. **IOS Developers Ship 29% Fewer Apps in 2017, the First Ever Decline - And More Trends to Watch**. 2018. Disponível em: <https://blog.appfigures.com/category/data-bits/>. Acesso em: 10 nov. 2018.

APPLE. **About Developing for Mac**. 2015. Disponível em: https://developer.apple.com/library/archive/documentation/MacOSX/Conceptual/OSX_Technology_Overview/About/About.html#/apple_ref/doc/uid/TP40001067-CH204-TPXREF101. Acesso em: 06 fev. 2019.

ARDUINO. **Arduino**. 2018. Disponível em: <https://www.arduino.cc/en/Guide/Introduction>. Acesso em: 05 jun. 2018.

_____. **Arduino Mega 2560**. 2019. Disponível em: <https://store.arduino.cc/arduino-mega-2560-rev3>. Acesso em: 14 out. 2018.

_____. **SoftwareSerial Library**. 2019. Disponível em: <https://www.arduino.cc/en/Reference/SoftwareSerial>. Acesso em: 29 maio 2019.

ASSOCIAÇÃO BRASILEIRA DOS CRIADORES DE SUÍNOS (Brasília). **Produção de Suínos: Teoria e Prática**. Brasília: Integrall, 2014. 908 p. Disponível em: http://www.abcs.org.br/attachments/-01_Livro_producao_bloq.pdf. Acesso em: 16 set. 2018.

BILMAIA, Leonardo Tolentino. **Condicionamento de Sensores de Temperatura, Presença de Vazão e Corrente, Aplicados a um Sistema de Monitoramento de Energia de Aquecimento de Água via Placas Solares**. 2018. 98 f. TCC (Graduação) - Curso de Centro de Tecnologia e Urbanismo, Departamento de Engenharia Elétrica, Universidade Estadual de Londrina, Londrina, 2018. Disponível em: http://www.uel.br/ctu/deel/TCC/TCC2017_LeonardoTolentinoBilmaia.pdf. Acesso em: 18 maio 2019.

BORGES, Pedro Hurtado de Mendoza *et al.* Sistema Automatizado de Baixo Custo para Produtores Rurais: controle e monitoramento do ambiente térmico na suinocultura. **Revista Eletrônica Competências Digitais Para Agricultura Familiar**, Mato Grosso, v. 4, n. 2, fev. 2018. Disponível em: <http://codaf.tupa.unesp.br:8082/index.php/recodaf/article/view/79/165>. Acesso em: 30 maio 2019.

BORTOLOZZO, Fernando Pandolfo *et al.* **Estratégias de redução do catabolismo lactacional manejando a ambiência na maternidade**. 2010. 12 f. Monografia (Especialização) - Curso de Faculdade de Veterinária, Departamento de Medicina Animal, Universidade Federal do Rio Grande do Sul, Porto Alegre, 2010. Disponível em: <https://www.yumpu.com/pt/document/view/5513889/estrategias-de-reducao-do-catabolismo-lactacional-suinotec>. Acesso em: 23 ago. 2018.

CASTRO, Fabrício Faleiros de. **ZOOTECNIA I (Suínos)**. 2016. 18 f. Monografia (Especialização) - Curso de Faculdade de Ciências Agrárias e Veterinárias, Departamento de Zootecnia, Unesp, São Paulo, 2016. Disponível em: http://www.fcav.unesp.br/Home/departamentos/zootecnia/edneypereiradasilva/aula-2_prof._fabricio.pdf. Acesso em: 02 nov. 2018.

CECCHIN, Daiane. **AMBIENTE TERMOACÚSTICO, QUALIDADE DO AR E BEM-ESTAR DE SUÍNOS EM INSTALAÇÕES COM DIFERENTES TIPOLOGIAS DE BAIAS**. 2016. 132 f. Tese (Doutorado) - Curso de Engenharia Agrícola, Universidade Federal de Lavras, Lavras, 2016. Disponível em: http://repositorio.ufla.br/bitstream/1/11513/2/TESE_Ambiente%20termoac%C3%BAstico%2C%20qualidade%20do%20ar%20e%20bem-estar%20de%20su%C3%ADnos%20em%20instala%C3%A7%C3%B5es%20com%20diferentes%20tipologias%20de%20bais.pdf. Acesso em: 09 set. 2018.

COSTA, Filipe Antonio dalla; COSTA, Osmar Antonio dalla. O Bem-estar de Suínos como Estratégia para Agregação de Valor. In: X CONGRESSO NORDESTINO DE PRODUÇÃO ANIMAL, 10, 2015, Teresina. **O Bem-estar de Suínos como Estratégia para Agregação de Valor**. Teresina: Snpa, 2015. p. 1 - 3. Disponível em: <https://ainfo.cnptia.embrapa.br/digital/bitstream/item/143313/1/final8031.pdf>. Acesso em: 15 set. 2018.

COUTINHO, Gabriel Santos *et al.* **Conforto térmico e manejo de suínos na maternidade levando em consideração o bem-estar animal.** 2014. Disponível em: http://www.nutritime.com.br/arquivos_internos/artigos/ARTIGO232.pdf. Acesso em: 10 set. 2018.

DATASHEETCAFE. **KY-038 PDF Arduino – Microphone Sound Sensor Module.** 2016. Disponível em: <http://www.datasheetcafe.com/ky-038-pdf-arduino/>. Acesso em: 29 jan. 2019.

DEVMEDIA. **Aplicações Móveis: Nativas ou Web?** 2015. Disponível em: <https://www.devmedia.com.br/aplicacoes-moveis-nativas-ou-web/30392>. Acesso em: 10 nov. 2018.

_____. **Como funcionam as aplicações web.** 2012. Disponível em: <https://www.devmedia.com.br/como-funcionam-as-aplicacoes-web/25888>. Acesso em: 08 maio 2019.

_____. **Introdução ao TypeScript.** 2016. Disponível em: <https://www.devmedia.com.br/introducao-ao-typescript/36729>. Acesso em: 07 maio 2019.

_____. **O que é JSON.** 2011. Disponível em: <https://www.devmedia.com.br/o-que-e-json/23166>. Acesso em: 18 maio 2019.

EMBRAPA. **Estatísticas | Desempenho da produção.** 2019. Disponível em: <https://www.embrapa.br/suinos-e-aves/cias/estatisticas>. Acesso em: 01 jun. 2019.

_____. **Produção Suínos.** 2003. Disponível em: <http://www.cnpsa.embrapa.br/SP/suinos/>. Acesso em: 02 nov. 2018.

_____. **Qualidade da carne suína.** 2018. Disponível em: <https://www.embrapa.br/qualidade-da-carne/carne-suina>. Acesso em: 01 jun. 2018.

ESPRESSIF SYSTEMS. **ESPRESSIF SMART CONNECTIVITY PLATFORM: ESP8266.** 2013. Disponível em: https://cdn-shop.adafruit.com/datasheets/ESP8266_Specifications_English.pdf. Acesso em: 15 nov. 2018.

EVANS, Martin; NOBLE, Joshua; HOCHENBAUM, Jordan. **Arduino in action.** Shelter Island: Manning, 2013. 370 p.

EXCELLENT WEBWORLD. **Top 5 Best Hybrid Mobile App Frameworks for 2018 to 2019.** 2018. Disponível em: <https://www.excellentwebworld.com/top-5-hybrid-mobile-app-frameworks/>. Acesso em: 11 nov. 2018.

FARM ANIMAL WELFARE COUNCIL. **The Five Freedoms.** 2009. Disponível em: <https://www.gov.uk/government/publications/fawc-report-on-farm-animal-welfare-in-great-britain-past-present-and-future>. Acesso em: 04 out. 2018.

FEDERIZZI, Ketlen Cilmara. **Impacto do estresse térmico na produção de suínos**. 2017. Disponível em: <https://opresenterural.com.br/impacto-do-estresse-termico-na-producao-de-suinos/>. Acesso em: 01 jun. 2018.

FERREIRA, Jeferson Luiz Miranda. **SEGURANÇA EM REDES SEM FIO**. 2013. 49 f. TCC (Graduação) - Curso de Curso de Especialização em Configuração e Gerenciamento de Servidores e Equipamentos de Rede, Departamento Acadêmico de Eletrônica, Universidade Tecnológica Federal do Paraná, Curitiba, 2013. Disponível em: http://repositorio.roca.utfpr.edu.br/jspui/bitstream/1/2412/1/CT_GESER_IV_2014_03.pdf. Acesso em: 13 nov. 2018.

FINGERPOINTENG. **ESP 8266 ESP-01 WI-FI MODULE**. 2018. Disponível em: <https://www.fingerpointengg.com/product/esp-8266-esp-01-wifi-module/>. Acesso em: 29 jan. 2019.

FUTURE ELECTRONIC CORPORATION. **Bluetooth module HC-05**. Disponível em: http://www.fecegypt.com/uploads/dataSheet/1522241762_bluetooth%20module%20hc05.pdf. Acesso em: 15 out. 2018.

GARTNER. **Worldwide Smartphone Sales to End Users by Operating System in 2017**. 2018. Disponível em: <https://www.gartner.com/en/newsroom/press-releases/2018-02-22-gartner-says-worldwide-sales-of-smartphones-recorded-first-ever-decline-during-the-fourth-quarter-of-2017>. Acesso em: 04 nov. 2018.

GOK, Nizamettin; KHANNA, Nitin. **Building Hybrid Android Apps: with Java and JavaScript**. Sebastopol: O'reilly, 2013. 155 p. Disponível em: <https://doc.lagout.org/programming/Multi-Language/Building%20Hybrid%20Android%20Apps%20with%20Java%20and%20JavaScript%20%5BGok%20%26%20Khanna%202013-08-10%5D.pdf>. Acesso em: 11 nov. 2018.

GONÇALVES, Vinicius. **Criação De Suínos**. 2012. Disponível em: <https://novonegocio.com.br/criacoes/criacao-de-suinos/>. Acesso em: 31 out. 2018.

INSTITUTO NCB. **Como utilizar o sensor de temperatura e umidade relativa DHT22 com o NodeMCU (MIC162)**. Disponível em: <http://www.newtoncbraga.com.br/index.php/microcontrolador/143-tecnologia/14349-como-utilizar-o-sensor-de-temperatura-e-umidade-relativa-dht22-com-o-nodemcu-mic162>. Acesso em: 26 out. 2018.

IONIC. **Install the Ionic CLI**. 2019. Disponível em: <https://ionicframework.com/docs/installation/cli>. Acesso em: 25 abr. 2019.

_____, **What is Ionic Framework?** 2019. Disponível em: <https://ionicframework.com/docs/intro>. Acesso em: 31 maio 2019.

ISWEEK. **Humidity Sensitive Resistor - HR202L**. Disponível em: https://www.isweek.com/product/humidity-sensitive-resistor-hr202l_2691.html. Acesso em: 26 out. 2018.

KIEFER, Charles *et al.* Respostas de suínos em terminação mantidos em diferentes ambientes térmicos. **Revista Brasileira Saúde e Produção Animal**, Salvador, v. 11, n. 2, p.496-504, jun. 2010. Disponível em: <http://revistas.ufba.br/index.php/rbspa/article/view/1649/985>. Acesso em: 10 set. 2018.

LANDIM, Wikerson. **O que é Wi-Fi?** 2012. Disponível em: <https://www.tecmundo.com.br/wi-fi/197-o-que-e-wi-fi-.htm>. Acesso em: 15 nov. 2018.

LECHETA, Ricardo R. **Desenvolvendo para iPhone e iPad**: aprenda a desenvolver aplicações utilizando o iOS SDK. 5. ed. São Paulo: Novatec, 2017. 607 p.

_____, Ricardo R. **Google Android**: Aprenda a criar aplicações para dispositivos móveis com o Android SDK. 4. ed. São Paulo: Novatec, 2015. 1016 p.

LEE, Ho-chul; YEO, Hyun. Design and Implimentation of Pig Farm Monitoring System for Ubiquitous Agriculture. **IEEE**, Sunchon, 2010. Disponível em: <https://ieeexplore.ieee.org/document/5674770>. Acesso em: 05 out. 2018.

LEE, Wei-meng. **Android Application Development**. Indianapolis: Wiley, 2011. 450 p. Disponível em: <http://www.kmvportal.co.in/Course/MAD/Android%20Book.pdf>. Acesso em: 18 out. 2018.

MASTERWALKER ELETRONIC SHOP. **Sensor (Detector) de Som - KY-038**. Disponível em: <https://www.masterwalkershop.com.br/sensor-detector-de-som-ky-038>. Acesso em: 06 out. 2018.

MATOS, Tiago Melillo de. **AUTOMAÇÃO RESIDENCIAL DE BAIXO CUSTO UTILIZANDO ARDUINO**. 2017. 40 f. Monografia (Especialização) - Curso de Engenharia de Controle e Automação, Universidade Federal de Ouro Preto, Ouro Preto, 2017. Disponível em: http://www.monografias.ufop.br/bitstream/35400000/968/1/MONOGRAFIA_Automa%C3%A7%C3%A3oResidencialBaixo.pdf. Acesso em: 16 nov. 2018.

MAZUCHETTI, Muriel Rampinelli. **ESTUDO DE FRAMEWORK MULTIPLATAFORMA APLICADO AO DESENVOLVIMENTO DE UM PROTÓTIPO DE APLICATIVO MOBILE HÍBRIDO PARA O CONTROLE FINANCEIRO PESSOAL**. 2015. 79 f. TCC (Graduação) - Curso de Ciência da Computação, Universidade do Extremo Sul Catarinense, Criciúma, 2015.

MELLAGI, Ana Paula Gonçalves *et al.* **Aspectos nutricionais de matrizes suínas durante a lactação e o impacto na fertilidade**. 2010. 30 f. Monografia (Especialização) - Curso de Faculdade de Agronomia, Universidade Federal do Rio Grande do Sul, Bento Gonçalves, 2010. Disponível em: <https://www.researchgate.net/publication/272834956/download>. Acesso em: 22 set. 2018.

MONK, Simon. **30 Projetos com Arduino**. 2. ed. Porto Alegre: Bookman, 2014. 214 p.

_____, Simon. **Programação com Arduino: Começando com sketches**. Porto Alegre: Bookman, 2013. 148 p. Tradução de: Anatólio Laschuk.

MOUSER ELECTRONICS. **DHT11**. Disponível em: <https://www.mouser.com/ds/2/758/DHT11-Technical-Data-Sheet-Translated-Version-1143054.pdf>. Acesso em: 05 out. 2018.

MORETTI, Bernardo Alexandre. **Proposta de Sistema Embarcado Para o Auxílio e Monitoramento de Idosos**. 2015. 171 f. TCC (Graduação) - Curso de Engenharia Elétrica, Universidade de São Paulo, São Carlos, 2015. Disponível em: http://www.tcc.sc.usp.br/tce/disponiveis/18/180450/tce-16022016-181527/publico/Bernardo_Alexandre_Moretti_tcc.pdf. Acesso em: 25 maio 2019.

MYSQL. **MySQL Workbench**. 2019. Disponível em: <https://www.mysql.com/products/workbench>. Acesso em: 21 abr. 2019.

NASCIMENTO, Luiz Carlos do; BONETTI, Tiago Piperno. **DESENVOLVENDO APLICAÇÕES MULTIPLATAFORMA PARA DISPOSITIVOS MÓVEIS UTILIZANDO IONIC E PHONEGAP**. 2016. 5 f. TCC (Graduação) - Curso de Ciência da Computação, Unipar, Paranavaí, 2016. Disponível em: http://web.unipar.br/~seinpar/2016/publicacao/Luiz_Carlos_Junior.pdf. Acesso em: 10 nov. 2018.

ODDWIRES. **HC-05 BLUETOOTH TRANSCEIVER MODULE RS232 / TTL MASTER / SLAVE MODULE**. 2019. Disponível em: <https://www.oddwires.com/hc-05-bluetooth-transceiver-module-rs232-ttl-master-slave-module/>. Acesso em: 29 jan. 2019.

OLIVEIRA, Marcelo Eduardo de. **Desenvolvimento de sistema automatizado de monitoramento de ambientes de produção animal, utilizando uma rede de sensores sem fio**. 2015. 57 f. Dissertação (Mestrado) - Curso de Faculdade de Zootecnia e Engenharia de Alimentos, Universidade de São Paulo, Pirassununga, 2015. Disponível em: <https://www.teses.usp.br/teses/disponiveis/74/74134/tde-01062016-102252/publico/ME5582191COR.pdf>. Acesso em: 30 maio 2019.

OLIVEIRA, Marcos Vinícius Cardoso de; LIMA, Lucas Henrique. **SISTEMA DE COMUNICAÇÃO DE DADOS UTILIZANDO ARDUINO E MÓDULO RF 433 MHz**. 2017. Disponível em: <http://repositorio.uniceub.br/jspui/bitstream/235/11360/1/SisCom%20RF.pdf>. Acesso em: 16 nov. 2018.

PASSOS, Bruno Pereira. **SISTEMA BLUETOOTH PARA CONTROLE DE ACESSÓRIOS VEICULARES UTILIZANDO SMARTPHONE COM ANDROID**. 2011. 83 f. TCC (Graduação) - Curso de Engenharia de Computação, Centro Universitário de Brasília, Brasília, 2011. Disponível em: <https://repositorio.uniceub.br/jspui/bitstream/123456789/3132/2/20614551.pdf>. Acesso em: 20 nov. 2018.

PHP. **O que é o PHP?** 2019. Disponível em:
https://www.php.net/manual/pt_BR/intro-whatcando.php. Acesso em: 29 abr. 2019.

PILAR, Carina Ponzoni. **AVALIAÇÃO DAS ARQUITETURAS DE DESENVOLVIMENTO PARA DISPOSITIVOS MÓVEIS**. 2013. 89 f. TCC (Graduação) - Curso de Bacharelado em Sistemas de Informação, Centro de Computação e Tecnologia da Informação, Universidade de Caxias do Sul, Caxias do Sul, 2013. Disponível em:
<https://repositorio.ucs.br/xmlui/bitstream/handle/11338/1245/TCC%20Carina%20Ponzoni%20Pilar.pdf?sequence=1&isAllowed=y>. Acesso em: 20 nov. 2018.

PINTO, Josuel Alfredo Vilela. **MANEJO DA UMIDADE RELATIVA DO AR DURANTE A ARMAZENAMENTO E SUA RELAÇÃO COM O AMADURECIMENTO E DISTÚRBIOS FISIOLÓGICOS EM FRUTAS**. 2012. 94 f. Tese (Doutorado) - Curso de Pós-graduação em Agronomia, Universidade Federal de Santa Maria, Santa Maria, 2012. Disponível em:
<https://repositorio.ufsm.br/bitstream/handle/1/3210/PINTO%2C%20JOSUEL%20ALFREDO%20VILELA.pdf?sequence=1&isAllowed=y>. Acesso em: 22 set. 2018.

POUSO, Manuella Thereza Pereira. **SISTEMA DE AUTOMAÇÃO E CONTROLE DE UM SISTEMA DE IRRIGAÇÃO**. 2012. 85 f. TCC (Graduação) - Curso de Engenharia da Computação, Centro Universitário de Brasília, Brasília, 2012. Disponível em:
<https://www.passeidireto.com/arquivo/25766808/sistema-de-automacao-e-controle-de-um-sistema-de-irrigacao>. Acesso em: 04 nov. 2018.

PREZOTTO, Ezequiel Douglas; BONIATI, Bruno Batista. **Estudo de Frameworks Multiplataforma Para Desenvolvimento de Aplicações Mobile Híbridas**. 2014. 8 f. TCC (Graduação) - Curso de Tecnologia em Sistemas Para Internet, Universidade Federal de Santa Maria, Frederico Westphalen, 2014. Disponível em:
<http://www.eati.info/eati/2014/assets/anais/artigo8.pdf>. Acesso em: 10 nov. 2018.

ROBOCORE. **Comparativo Arduino**. 2017. Disponível em:
<https://www.robocore.net/tutoriais/comparativo-arduino.html>. Acesso em: 21 nov. 2018.

ROHM SEMICONDUCTOR. **Ambient Light Sensor IC Series: BH1750FVI**. 2011. Disponível em: <https://www.mouser.com/ds/2/348/bh1750fvi-e-186247.pdf>. Acesso em: 06 out. 2018.

ROHR, Stefan Alexander; COSTA, Osmar Antonio dalla; COSTA, Filipe Antônio dalla. **Bem-estar animal na produção de suínos**. Brasília: Abcs, 2016. 38 p. Disponível em: <http://www.agricultura.gov.br/assuntos/boas-praticas-e-bem-estar-animal/arquivos-publicacoes-bem-estar-animal/cartilha-embrapa-abcs-mapa-sebrae-bem-estar-na-granja.pdf>. Acesso em: 22 set. 2018.

SANTOS, Antônio Henrique dos; CARVALHO, Nelson Ribeiro. **Frameworks e seus Benefícios no Desenvolvimento de Software**. Disponível em:
http://revistapensar.com.br/tecnologia/pasta_upload/artigos/a95.pdf. Acesso em: 01 maio 2018.

SCHMIDT, Nádia Solange. **DEMANDAS ATUAIS E FUTURAS DA CADEIA PRODUTIVA DE SUÍNOS**. 2017. Disponível em: <https://www.embrapa.br/documents/1355242/0/CIAS+-+Agropensa+-+Demandas+atuais+e+futuras+da+cadeia+produtiva+de+su%C3%ADnos.pdf>. Acesso em: 09 set. 2018.

SILVA, Cristiane Rocha; JESUS, José Carlos dos Santos. **Análise da percepção do valor da informação e do benefício-custo da tecnologia da informação por suinocultores na região de Passos, MG**. 2007. Disponível em: <http://www.redalyc.org/html/878/87890302/>. Acesso em: 02 jun. 2018.

SILVA, Marcos Aurélio Saminez da. **Aferição e Monitoramento Remoto de Temperatura e Umidade em Datacenter**. 2018. 50 f. Monografia (Especialização) - Curso de Engenharia Elétrica, Departamento de Engenharia de Eletricidade, Universidade Federal do Maranhão, São Luís, 2018. Disponível em: <https://monografias.ufma.br/jspui/bitstream/123456789/2434/1/MARCOSAURELIOSILVA.pdf>. Acesso em: 31 maio 2019.

SONGLE RELAY. **Relay ISO9002**. 2016. Disponível em: <http://www.circuitbasics.com/wp-content/uploads/2015/11/SRD-05VDC-SL-C-Datasheet.pdf>. Acesso em: 28 maio 2019.

SOUZA, Denaildo Junio Borges de; VIANA, Felipe Cabral. **COMPARANDO A EFICÁCIA DE BANCO DE DADOS ORIENTADO A OBJETOS COM BANCO DE DADOS RELACIONAL ATRAVÉS DE PESQUISA BIBLIOGRÁFICA E IMPLEMENTAÇÃO DE PROTÓTIPO**. 2016. 51 f. Monografia (Especialização) - Curso de Bacharelado em Sistemas de Informação, Instituto Federal de Educação, Ciência e Tecnologia Fluminense, Campos dos Goytacazes, 2016. Disponível em: <http://bd.centro.iff.edu.br/bitstream/123456789/1186/1/Trabalho%20de%20Conclus%C3%A3o%20de%20Curso.pdf>. Acesso em: 27 abr. 2019.

SOUZA, Fábio. **Arduino MEGA 2560**. 2014. Disponível em: <https://www.embarcados.com.br/arduino-mega-2560/>. Acesso em: 03 nov. 2018.

SOUZA, Francine Aparecida. **Análise de fatores ambientais na criação de suínos em camas sobrepostas e produção de biogás**. 2014. 122 f. Tese (Doutorado) - Curso de Análise de Fatores Ambientais na Criação de Suínos em Camas Sobrepostas e Produção de Biogás, Universidade Federal de Lavras, Lavras, 2014. Disponível em: <http://repositorio.ufla.br/bitstream/1/2970/1/TESE%20An%C3%A1lise%20de%20fatores%20ambientais%20na%20cria%C3%A7%C3%A3o%20de%20su%C3%ADnos%20em%20camas%20sobrepostas%20e%20produ%C3%A7%C3%A3o%20de%20biog%C3%A1s.pdf>. Acesso em: 15 set. 2018.

STACKEXCHANGE. **How to connect an ESP8266 using an ESP-01 adapter to an Arduino Uno**. 2018. Disponível em: <https://arduino.stackexchange.com/questions/54348/how-to-connect-an-esp8266-using-an-esp-01-adapter-to-an-arduino-uno>. Acesso em: 28 maio 2019.

SUNROM ELECTRONICS / TECHNOLOGIES. **Digital Light Sensor - BH1750FVI**. Disponível em: <https://www.sunrom.com/p/digital-light-sensor-bh1750fvi>. Acesso em: 26 out. 2018.

_____. **Light Dependent Resistor - LDR**. 2008. Disponível em: <https://www.sunrom.com/get/443700>. Acesso em: 06 out. 2018.

TAVARES, Henrique Leal. **Introdução a desenvolvimento de aplicações híbridas: Curso de Tecnologia em Análise e Desenvolvimento de Sistemas**. 2016. Disponível em: <http://revista.fatecgarca.edu.br/index.php/efatec/article/view/113/109>. Acesso em: 13 maio 2018.

TELECO. **Estatísticas de Celulares no Brasil**. 2018. Disponível em: <http://www.teleco.com.br/ncel.asp>. Acesso em: 01 out. 2018.

TEXAS INSTRUMENTS. **LM35 Precision Centigrade Temperature Sensors**. 2017. Disponível em: <http://www.ti.com/lit/ds/symlink/lm35.pdf>. Acesso em: 04 out. 2018.

THOMAZINI, Daniel; ALBUQUERQUE, Pedro Urbano de. **Sensores Industriais: Fundamentos e Aplicações**. São Paulo: Érica, 2005. 220 p.

WEBSOPTIMIZATION. **Top 5 hybrid Mobile App Frameworks in 2018 - Choose the best one for you**. 2018. Disponível em: <https://www.weboptimization.com/blog/hybrid-mobile-app-frameworks/>. Acesso em: 11 nov. 2018.

WENDLING, Marcelo; **Sensores**. 2010. Disponível em: <http://www2.feg.unesp.br/Home/PaginasPessoais/ProfMarceloWendling/4---sensores-v2.0.pdf>. Acesso em: 16 nov. 2018.

WILKEN, Jeremy. **Ionic in Action: HYBRID MOBILE APPS WITH IONIC AND ANGULARJS**. Shelter Island: Manning, 2016. 282 p. Disponível em: <http://pepa.holla.cz/wp-content/uploads/2016/12/Ionic-in-Action.pdf>. Acesso em: 10 nov. 2018.

APÊNDICE(S)

APÊNDICE A - INSTALAÇÃO DO IONIC

De acordo com as informações repassadas pela documentação do site oficial do *framework*, para a sua instalação faz-se necessário primeiramente o download e instalação das ferramentas Node.js e npm.

Após estas duas estarem devidamente instaladas e configuradas, torna-se possível a realização do Ionic via linha de comando, conforme ilustrado na figura 1.

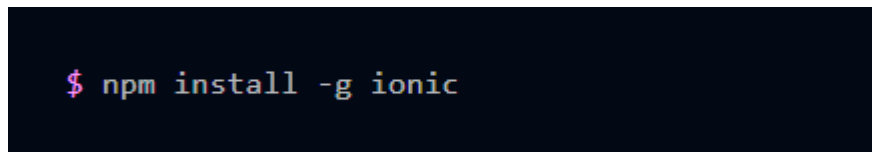
A screenshot of a terminal window with a dark background. It shows a command prompt symbol (\$) followed by the text 'npm install -g ionic' in a light blue monospace font.

Figura 1. Comando de instalação do Ionic.

Posteriormente faz-se necessário algumas configurações de acordo com cada plataforma que se deseja rodar a aplicação, bem como o download e instalação do SDK do sistema operacional de cada uma delas.

APÊNDICE B - ARTIGO CIENTÍFICO

Automatização e Monitoramento de Processos Relacionados ao Bem-Estar Animal Dentro da Cadeia Produtiva da Suinocultura por Meio de Microcontroladores Integrados a uma Aplicação Mobile

João Paulo Niehues Wessler¹, Paulo João Martins²

¹Acadêmico do Curso de Ciência da Computação - Unidade Acadêmica de Ciências, Engenharias e Tecnologias - Universidade do Extremo Sul Catarinense (UNESC) - Criciúma – SC

²Professor do Curso de Ciência da Computação - Unidade Acadêmica de Ciências, Engenharias e Tecnologias - Universidade do Extremo Sul Catarinense (UNESC) - Criciúma - SC

joaopaulowessler@gmail.com, pjm@unesc.net

Abstract. *The use of technology is being more present in production environments, in the scenario of pig production is no different, where the continuous technological implementation, increasingly provides increased productivity and assistance in the management of animals. Given this context, the present work aims at the research and development of a prototype for the monitoring of variables related to animal welfare levels in swine. For that, the Arduino microcontroller was integrated with sensors specific to each of the variables, through which the data is collected and sent to a server in the cloud. Later for the management of this information, the development of a mobile application was done, using the framework Ionic.*

Resumo. *O uso da tecnologia está cada vez mais presente nos ambientes de produção, onde no cenário de criação de suínos não é diferente, onde a contínua implementação tecnológica, proporciona cada vez mais o aumento da produtividade e o auxílio no manejo dos animais. Diante deste contexto, o presente trabalho visa a pesquisa e o desenvolvimento de um protótipo para o monitoramento de variáveis relacionadas aos níveis de bem-estar animal na suinocultura. Para tal, utilizou-se do microcontrolador Arduino integrado a sensores específicos para cada uma das variáveis, ao qual por intermédio deste, os dados são coletados e enviados para um servidor na nuvem. Posteriormente para o gerenciamento dessas informações, foi realizado o desenvolvimento de uma aplicação mobile, utilizando-se do framework Ionic.*

1. Introdução

Segundo informações da Empresa Brasileira de Pesquisa Agropecuária (Embrapa) (2019), o Brasil é o quarto maior produtor e exportador de carne suína do mundo, chegando em 2018 a 3,76 milhões de toneladas produzidas e a 730 mil toneladas exportadas. Desempenho proporcionado principalmente por mudanças organizacionais e o contínuo incremento tecnológico na produção.

A suinocultura brasileira está posicionada entre as cadeias produtivas mais avançadas do mundo. O uso de alta tecnologia nas áreas de nutrição, genética, manejo e instalações, permitiu a produção de carne com elevados padrões de qualidade (SCHMIDT, 2017).

Com a implementação da tecnologia na cadeia produtiva, além de alavancar a produção, gerou-se também melhores índices de conforto ao animal, a qual segundo Coutinho et al. (2014), o bem-estar animal é o estado de harmonia entre o animal e seu ambiente, caracterizado por condições físicas e fisiológicas ideais a qualidade de vida do mesmo.

Fator este que representa também um papel crucial no nível de perdas durante a produção, fazendo com que os produtores se adaptem as mudanças nas regras de qualidade de vida animal, regulamentos e boas práticas de manejo (COSTA; COSTA, 2015).

Dentro dos padrões de bem-estar animal da suinocultura, para que se possa ter circunstâncias ideais para um bom desenvolvimento, diversas são as variáveis a serem tratadas pelos suinocultores, dentre elas: umidade relativa, temperatura, frequência sonora e luminosidade.

Os suínos são animais homeotérmicos, pois mantêm sua temperatura interna constante e trocam calor com o ambiente. Entretanto, para casos onde a boa conduta no manejo dos animais não é respeitada, o desenvolvimento e a produtividade esperada pelos mesmos diminuem. Fator gerado principalmente pelo estresse provocado por um ambiente não confortável aos mesmos, com isso gerando inquietação e uma menor conversão alimentar, acarretando grandes prejuízos aos produtores (FEDERIZZI, 2017).

A fim de manter níveis climáticos favoráveis ao bom desenvolvimento dos animais, atualmente os produtores dispõem-se de equipamentos como ventiladores, umidificadores e desumidificadores em suas granjas. Porém, como se trata de um procedimento manual, requer mão de obra qualificada e vigilância constante, ao qual muitas vezes geram transtornos e consequentemente afetam a qualidade e a produtividade dos suínos.

Com o intuito de proporcionar melhorias neste processo, algumas tecnologias estão disponíveis no mercado, dentre elas, pode ser citada a implementação de microcontroladores e sensores, a fim de possibilitar o controle automático dos equipamentos citados anteriormente, desta forma, reduzindo a margem de erros operacionais e garantindo boas condições de vida aos animais.

Além da implementação de microcontroladores e dos variados sensores para as situações que se deseja monitorar, torna-se indispensável por parte dos produtores, formas de visualizar os dados coletados de maneira ágil e prática, para o auxílio do manejo e o gerenciamento dos locais de produção.

Com base nisso, uma boa opção a ser utilizada são os dispositivos móveis, visto que, cada vez mais estes aparelhos estão no cotidiano das pessoas, devido principalmente à sua ótima portabilidade e usabilidade. Com base nisso, torna-se possível o desenvolvimento de uma aplicação móvel, para a visualização dos dados obtidos pelos microcontroladores e sensores.

No referente trabalho, propõe-se a utilização de microcontroladores no desenvolvimento de um protótipo, para o monitoramento e automatização de processos relacionados ao bem-estar animal, na cadeia produtiva da suinocultura (temperatura, umidade relativa do ar, frequência sonora e luminosidade), integrando as informações coletadas através de uma aplicação mobile, a fim de facilitar a visualização dos dados pelos suinocultores, com isso, agilizando a tomada de decisões e auxiliando no manejo, gerando assim uma maior produtividade e melhores índices de bem-estar animal.

2. Microcontroladores e Sensores

Microcontroladores podem ser entendidos como a junção de um microprocessador com interfaces de entrada e saída e outros periféricos, em apenas um único componente (POUSO, 2012).

O microprocessador, ou também conhecido por Unidade Central de Processamento, do inglês Central Processing Unit (CPU), tem como objetivo principal processar dados, buscando instruções na memória, decodificando e as executando. Já as interfaces de entrada e saída, disponibilizam a possibilidade de conexão com dispositivos externos como sensores, relés, entre outros (POUSO, 2012).

Já os sensores são dispositivos que de alguma forma são sensíveis aos tipos de energia presentes na natureza, como por exemplo a cinética, luminosa, térmica, e outras, e através delas relacionam as informações sobre uma grandeza previamente medida, como: aceleração, corrente, pressão, temperatura, umidade, velocidade, entre outras (THOMAZINI; ALBUQUERQUE, 2005).

A fim de realizar o controle de algumas variáveis relacionadas ao bem-estar dos animais na cadeia produtiva da suinocultura, pode ser encontrada uma diversificada variedade de equipamentos disponíveis no mercado, dentre eles o microcontrolador Arduino, que disponibiliza a capacidade de conexão com sensores externos, como os de temperatura, umidade, sonoridade e luminosidade.

2.1 Arduino

É um pequeno microcontrolador que possui vários pinos de conexão, a qual permite que outros circuitos eletrônicos externos sejam conectados a ele, como por exemplo, motores, relés, sensores, alto-falantes, entre outros. Também possui uma porta Universal Serial Bus (USB), que o permite se conectar a um computador com intuito de realizar o controle do fluxo de dados (MONK, 2014).

De acordo com o site oficial da plataforma (Arduino, 2019, tradução nossa), as principais vantagens são: custo, ambiente de programação simples e claro, multiplataforma e software / hardware open source e extensível.

3. Aplicações Híbridas

Aplicações híbridas ou também conhecidas por multiplataforma, possuem como principal objetivo, a capacidade de serem executadas em diversas plataformas, sem a necessidade de alteração do código fonte dos aplicativos. Sua estrutura é formada pela junção das metodologias *web apps* e nativa.

Diferentemente dos *web apps*, os aplicativos híbridos ficam instalados no dispositivo, tendo a possibilidade de operar em modo *off-line*, como também, podendo ser disponibilizados nas lojas de aplicativos, similar aos nativos. Normalmente são desenvolvidos em linguagem *web*, sendo interpretados pelo *browser* do SO em que estão contidos (PREZOTTO; BONIATI, 2014).

De acordo com Gok e Khanna (2013, tradução nossa), os aplicativos híbridos desempenham um papel fundamental na redução da lacuna existente entre os recursos dos navegadores e dos dispositivos, permitindo desta forma, que os desenvolvedores possam aproveitar o melhor dos dois mundos, ao qual geralmente utilizam-se de combinações de HTML5, CSS, JavaScript juntamente com os SDKs específicos de cada plataforma, como o Java no Android e o Objective-C para o IOS.

Dentre os colocados no top cinco dos mais utilizados pelos desenvolvedores no ano de 2018, segundo Excellent Webworld (2018, tradução nossa) e Websoptimization (2018, tradução nossa), está o Ionic.

3.1 FrameWork Ionic

Este *framework* pode ser entendido, como a combinação de tecnologias como HTML, CSS e JavaScript, que foram projetadas com intuito de facilitar e agilizar a construção de aplicativos híbridos com destaque visual. Contudo, para que ele possa ser executado, outras ferramentas devem acompanhá-lo, sendo elas, o Angular como framework de aplicação *web* JavaScript e o Cordova sendo encarregado pela construção e interpretação do aplicativo junto ao SO (MAZUCHETTI, 2015).

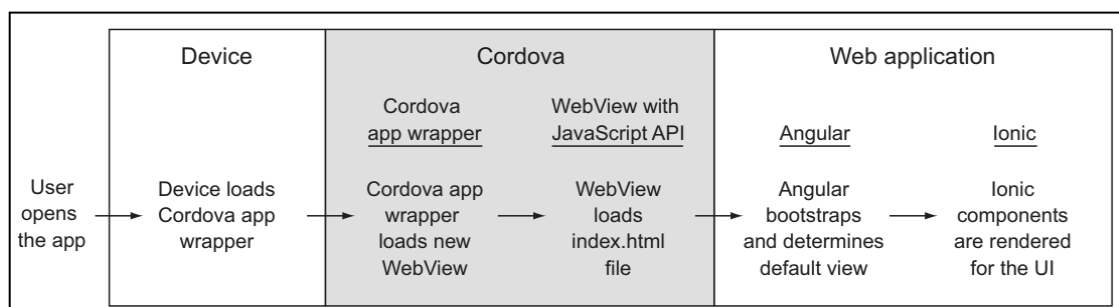


Figura 1. Resumo de ferramentas necessários para a execução do Ionic

Conforme pode ser observado na figura 1, o primeiro quadro corresponde ao dispositivo, que detêm o sistema operacional. Já o segundo retrata o Cordova, que tem por finalidade a interpretação entre o SO e o aplicativo, criando uma aplicação nativa que é composta por um WebView, que possui características de uma página web, onde a aplicação é internamente executada (WILKEN, 2016, tradução nossa).

Uma outra característica marcante do Cordova é que o faz ter uma essencial utilização junto ao Ionic, é o fato do mesmo possibilitar o desenvolvimento de aplicativos móveis, sem ter qualquer necessidade de programação de códigos nativos, como Java para o Android e Objective-C para o IOS, com isso possibilitando o desenvolvimento inteiramente utilizando-se de tecnologias web (MAZUCHETTI, 2015).

Já o último estágio é representando pelo Angular e o próprio Ionic, que tem como objetivos controlar a lógicas e os dados das aplicações.

4. Metodologia

Para a escolha de quais variáveis de monitoração seriam compreendidas no projeto, baseou-se em pesquisas realizadas em materiais de produção de suínos fornecidos pelo EMBRAPA e ABCS, ao qual avaliou-se quais as possibilidades que estavam em pauta e que contêm importância fundamental na cadeia produtiva dos suínos. Como também, a disponibilidade de sensores para o monitoramento e que fossem passíveis de integração com o microcontrolador selecionado.

As demais etapas estão baseadas no desenvolvimento do protótipo, iniciando-se primeiramente pelo levantamento dos recursos necessários, após isso estipulou-se o fluxo de dados e a modelagem do banco de dados, seguidos pela configuração do servidor responsável pelo armazenamento dos dados, montagem da simulação com a estruturação do microcontrolador e demais dispositivos, desenvolvimento da aplicação e por fim os testes de comunicação dos dados entre todas estas etapas e funcionalidades gerais do projeto.

4.1. Fluxo dos dados e modelo de comunicação

O diagrama de fluxo de dados estipulado (figura 2), é formado pelo mapeamento de todos os processos que o projeto é contido e todas as interconexões necessárias entre os mesmos. Com isso, possibilitando uma visão global, facilitando a percepção de todas as etapas a serem realizadas.

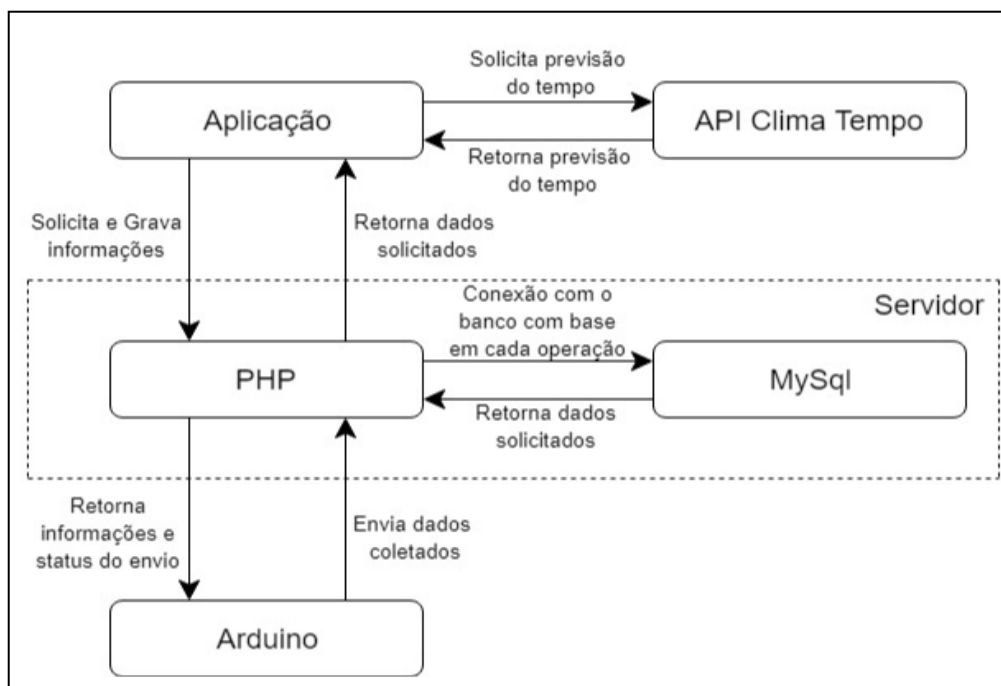


Figura 2. Fluxo de dados do projeto

O fluxo inicial estipulado, acontece no Arduino, onde este deve reunir as informações necessárias e realizar o envio para o banco de dados no servidor na nuvem. Sendo que, esta conexão é intermediada por um arquivo PHP localizado no próprio servidor, tendo este o objetivo de centralizar todas as operações que envolvem de alguma forma o uso da base de dados.

Por fim, a aplicação é a última parte do fluxo, ao qual nesta será possível realizar o gerenciamento das informações coletadas. Onde para a conexão com a base de dados, segue o mesmo formato citado anteriormente via arquivo PHP. Esta etapa também possui, a comunicação com uma API externa para a obtenção de previsão do tempo, sendo uma funcionalidade extra do projeto.

Com relação ao formato de comunicação entre todas estas etapas, se deu por meio de requisições HTTP, utilizando-se dos métodos GET e SET dependendo de cada ocasião, ao qual por meio destes são transmitidos os dados no formato JSON.

4.2. Montagem da simulação

O primeiro passo efetuado nesta etapa, foi propriamente a montagem física da maquete, ao qual foram acoplados em um suporte todos os equipamentos necessários, com isso, realizando as interconexões necessárias entre os mesmos, conforme o esquema ilustrado na figura 3.

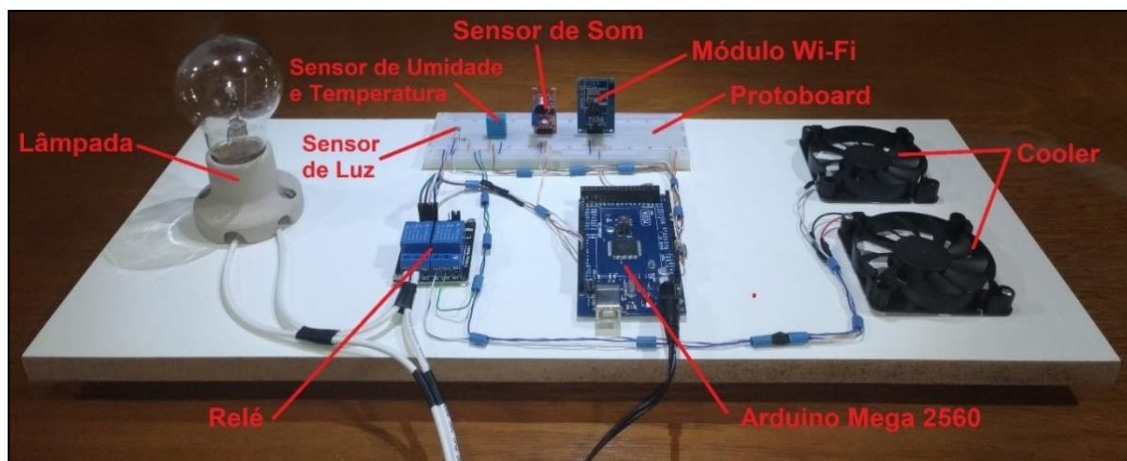


Figura 3. Simulação

Dentre os equipamentos utilizados, iniciou-se a montagem pelo protoboard, tendo este a responsabilidade de ser o suporte para a conexão entre os sensores e o microcontrolador.

A próxima etapa se deu na configuração dos sensores utilizados para a coleta das variáveis de bem-estar animal, tendo como primeiro deles o DHT11, que possui objetivo de verificar a temperatura e umidade. Com relação a captura dos dados de luminosidade, utilizou-se do sensor LDR e para a sonoridade, utilizou-se do KY-038.

A fim de realizar o controle e centralizar todos os dados coletados por cada um dos sensores citados anteriormente, utilizou-se do microcontrolador Arduino, em específico o modelo Mega 2560.

Dentre as principais vantagens do Arduino, que nortearam a escolha pela sua utilização no projeto, estão no baixo custo de aquisição se comparado com outras plataformas, software de código aberto e extensível, ambiente de programação de fácil acesso e uso, multiplataforma, ao qual pode ser executado com Windows, Macintosh OS X e o Linux e principalmente pelo fato de que as aplicabilidades do Arduino são basicamente ilimitadas, dependendo apenas do empenho e criatividade aplicada pelos desenvolvedores

Após o término do acoplamento do microcontrolador e dos sensores responsáveis pela coleta dos dados, a próxima etapa constitui-se na utilização e nas configurações necessárias do equipamento responsável pelo envio das informações coletadas para o servidor na nuvem, ao qual para esta função, foi escolhido o módulo Wi-Fi ESP8266 ESP-01.

Por fim foram instalados os equipamentos incumbidos de auxiliar no controle das variáveis de bem-estar, sendo a lâmpada simulando o aquecimento e os coolers o resfriamento.

4.3. Desenvolvimento da aplicação

Para o desenvolvimento da aplicação, utilizou-se do framework Ionic, ao qual teve como principais motivos para a sua escolha a possibilidade de criar aplicações móveis utilizando-se dos recursos de tecnologias web, como *HTML*, *CSS* e *JavaScript*, tendo como resultado uma aplicação que se comporta como nativa, ao qual pode desfrutar de todos os recursos do dispositivo.

Outro fator relevante para esta escolha, se dá pelo fato do Ionic possibilitar o desenvolvimento híbrido ou multiplataforma, ao qual, reduz consideravelmente o tempo de produção da aplicação, visto que, não se faz necessário ter conhecimentos específicos voltados para determinado sistema operacional.

Já com relação ao *framework* de *front-end* integrado ao Ionic, utilizado no projeto, foi o Angular, sendo este, uma plataforma utilizada para a construção de interfaces de aplicações desenvolvida pela Google, ao qual utiliza-se de tecnologias como *HTML*, *CSS* e *JavaScript*.

Para a criação de aplicações com Angular, pode-se usar as linguagens *JavaScript*, *TypeScript* e *Dart*. Porém como o Angular foi desenvolvido em *TypeScript*, a popularização desta, ficou maior junto ao *framework* em relação as demais. Em função disso, para o desenvolvimento deste projeto utilizou-se dele.

4.4. Funcionamento da aplicação

Após o término da etapa de desenvolvimento da aplicação, a mesma ficou composta por um conjunto de páginas, cada qual com suas finalidades específicas. Dentre estas, existem telas de consultas dos dados e de manutenção, sendo que todas elas são organizadas pelo conceito de menu, onde a partir deste é possível acessar todos os recursos do aplicativo.

Baseando-se em uma lógica para o acesso cronológico das telas na aplicação, tem-se na primeira página a de monitoramento sendo a principal delas, visto que é nesta que o usuário poderá acompanhar os últimos valores de cada sensor obtido pelo microcontrolador, com sua respectiva data e hora, e se julgar necessário, poderá realizar o acionamento dos equipamentos disponíveis, como também visualizar um histórico detalhado de todas as coletas realizadas. A figura 4 ilustra as funcionalidades mencionadas.

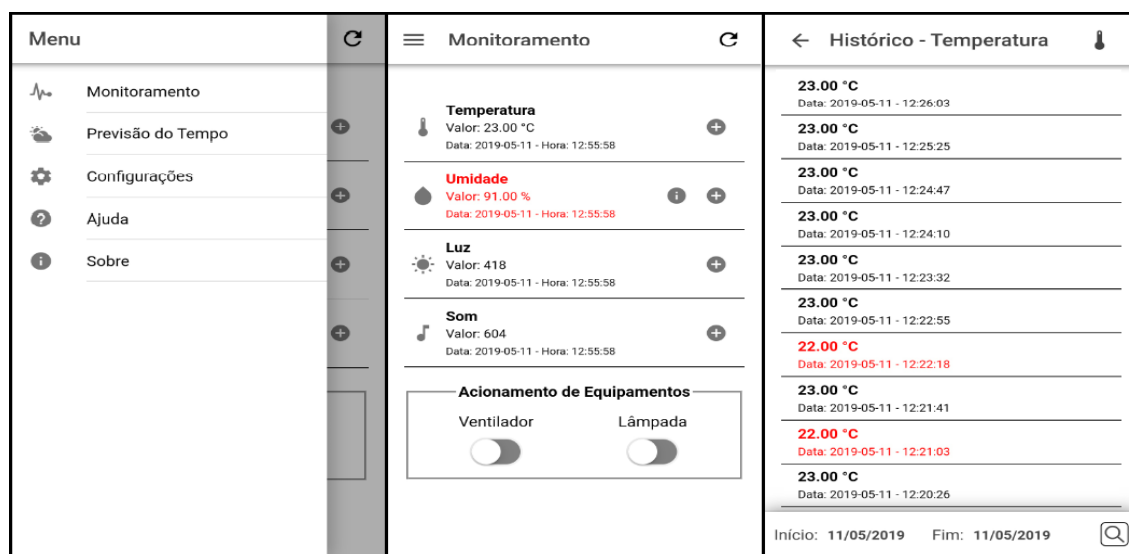


Figura 4. Telas do aplicativo

As demais funcionalidades consistem nas telas de previsão do tempo, configuração, consulta de logs, ajuda e sobre. A figura 5 representas as principais.



Figura 5. Telas do aplicativo

7. Conclusão

Com o término do trabalho em questão, além proporcionar a aquisição de conhecimento relacionado as tecnologias utilizadas para a realização do protótipo, também forneceu noções básicas referente aos conceitos relacionados a produção de suínos, ao qual pode-se compreender a importância que o manejo realizado corretamente, referente as variáveis de monitoração selecionadas, tem como influência no resultado final da produtividade dos animais.

Através do uso do microcontrolador Arduino juntamente com os demais dispositivos acoplados, tornou-se possível realizar o monitoramento das variáveis de ambiente de temperatura, umidade, luminosidade e sonoridade, obtendo-se êxito na realização da coleta das informações e envio para o servidor na nuvem, tendo apenas como empecilho os dados obtidos de frequência sonora, onde não se mostraram atrativos para um real monitoramento, visto que, não conseguiu-se obter com o sensor utilizado, dados que possam representar uma variação sonora do ambiente, ao qual todas as informações coletadas possuem valores constantes de acordo com o que é regulado no potenciômetro do equipamento.

Referente ao processo de desenvolvimento da aplicação, o mesmo possibilitou a aquisição de conhecimento à cerca do conceito de aplicações híbridas, realizadas por meio de *frameworks* multiplataformas, em específico o Ionic, onde foram elencadas as principais funcionalidades e benefícios que o mesmo disponibiliza, afim de ser implementado no projeto.

Com isso, foi possível obter como resultado, um aplicativo capaz de realizar os procedimentos propostos no projeto, utilizando-se de métodos ágeis e dinâmicos de programação fornecidos pelo Ionic, tendo estes, as características principais voltadas inteiramente para a utilização de tecnologias de desenvolvimento web.

Com relação ao projeto desenvolvido como um todo e com base nos testes realizados, obteve-se um resultado satisfatório quanto a integração entre todas as tecnologias implementadas, tendo como desfecho um protótipo de sistema de monitoramento de variáveis de bem-estar animal, que pode ser avaliado para a aplicação em um ambiente de produção de suínos, afim de otimizar e gerar melhores índices de produtividade.

Referências

- COSTA, Filipe Antonio dalla; COSTA, Osmar Antonio dalla. O Bem-estar de Suínos como Estratégia para Agregação de Valor. In: X CONGRESSO NORDESTINO DE PRODUÇÃO ANIMAL, 10, 2015, Teresina. O Bem-estar de Suínos como Estratégia para Agregação de Valor. Teresina: Snpa, 2015. p. 1 - 3. Disponível em: <https://ainfo.cnptia.embrapa.br/digital/bitstream/item/143313/1/final8031.pdf>. Acesso em: 15 set. 2018.
- COUTINHO, Gabriel Santos et al. Conforto térmico e manejo de suínos na maternidade levando em consideração o bem-estar animal. 2014. Disponível em: http://www.nutritime.com.br/arquivos_internos/artigos/ARTIGO232.pdf. Acesso em: 10 set. 2018.
- EMBRAPA. Estatísticas | Desempenho da produção. 2019. Disponível em: <https://www.embrapa.br/suinos-e-aves/cias/estatisticas>. Acesso em: 01 jun. 2019.
- EXCELLENT WEBWORLD. Top 5 Best Hybrid Mobile App Frameworks for 2018 to 2019. 2018. Disponível em: <https://www.excellentwebworld.com/top-5-hybrid-mobile-app-frameworks/>. Acesso em: 11 nov. 2018.
- FEDERIZZI, Ketlen Cilmar. Impacto do estresse térmico na produção de suínos. 2017. Disponível em: <https://opresenterural.com.br/impacto-do-estresse-termico-na-producao-de-suinos/>. Acesso em: 01 jun. 2018.
- GOK, Nizamettin; KHANNA, Nitin. Building Hybrid Android Apps: with Java and JavaScript. Sebastopol: O'reilly, 2013. 155 p. Disponível em: <https://doc.lagout.org/programming/Multi-Language/Building%20Hybrid%20Android%20Apps%20with%20Java%20and%20JavaScript%20%5BGok%20%26%20Khanna%202013-08-10%5D.pdf>. Acesso em: 11 nov. 2018.
- MAZUCHETTI, Muriel Rampinelli. ESTUDO DE FRAMEWORK MULTIPLATAFORMA APLICADO AO DESENVOLVIMENTO DE UM PROTÓTIPO DE APLICATIVO MOBILE HÍBRIDO PARA O CONTROLE FINANCEIRO PESSOAL. 2015. 79 f. TCC (Graduação) - Curso de Ciência da Computação, Universidade do Extremo Sul Catarinense, Criciúma, 2015.
- MONK, Simon. 30 Projetos com Arduino. 2. ed. Porto Alegre: Bookman, 2014. 214 p.
- POUSO, Manuella Thereza Pereira. SISTEMA DE AUTOMAÇÃO E CONTROLE DE UM SISTEMA DE IRRIGAÇÃO. 2012. 85 f. TCC (Graduação) - Curso de Engenharia da Computação, Centro Universitário de Brasília, Brasília, 2012. Disponível em: <https://www.passeidireto.com/arquivo/25766808/sistema-de-automacao-e-controle-de-um-sistema-de-irrigacao>. Acesso em: 04 nov. 2018.
- PREZOTTO, Ezequiel Douglas; BONIATI, Bruno Batista. Estudo de Frameworks Multiplataforma Para Desenvolvimento de Aplicações Mobile Híbridas. 2014. 8 f. TCC (Graduação) - Curso de Tecnologia em Sistemas Para Internet, Universidade Federal de Santa Maria, Frederico Westphalen, 2014. Disponível em: <http://www.eati.info/eati/2014/assets/anais/artigo8.pdf>. Acesso em: 10 nov. 2018.

THOMAZINI, Daniel; ALBUQUERQUE, Pedro Urbano de. Sensores Industriais: Fundamentos e Aplicações. São Paulo: Érica, 2005. 220 p.

WEBSOPTIMIZATION. Top 5 hybrid Mobile App Frameworks in 2018 - Choose the best one for you. 2018. Disponível em: <https://www.weboptimization.com/blog/hybrid-mobile-app-frameworks/>. Acesso em: 11 nov. 2018.

WILKEN, Jeremy. Ionic in Action: HYBRID MOBILE APPS WITH IONIC AND ANGULARJS. Shelter Island: Manning, 2016. 282 p. Disponível em: <http://pepa.holla.cz/wp-content/uploads/2016/12/Ionic-in-Action.pdf>. Acesso em: 10 nov. 2018.